Lecture Notes in Computer Science 3045

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA
Josef Kittler
University of Surrey, Guildford, UK
Jon M. Kleinberg
Cornell University, Ithaca, NY, USA
Friedemann Mattern
ETH Zurich, Switzerland
John C. Mitchell
Stanford University, CA, USA
Oscar Nierstrasz
University of Bern, Switzerland
C. Pandu Rangan
Indian Institute of Technology, Madras, India
Bernhard Steffen
University of Dortmund, Germany
Madhu Sudan
Massachusetts Institute of Technology, MA, USA
Demetri Terzopoulos
New York University, NY, USA
Doug Tygar
University of California, Berkeley, CA, USA
Moshe Y. Vardi
Rice University, Houston, TX, USA
Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Antonio Laganà Marina L. Gavrilova
Vipin Kumar Youngsong Mun
C.J. Kenneth Tan Osvaldo Gervasi (Eds.)

Computational Science and Its Applications – ICCSA 2004

International Conference
Assisi, Italy, May 14-17, 2004
Proceedings, Part III

Springer

Volume Editors

Antonio Laganà
University of Perugia, Department of Chemistry
Via Elce di Sotto, 8, 06123 Perugia, Italy
E-mail: lag@unipg.it

Marina L. Gavrilova
University of Calgary, Department of Computer Science
2500 University Dr. N.W., Calgary, AB, T2N 1N4, Canada
E-mail: marina@cpsc.ucalgary.ca

Vipin Kumar
University of Minnesota, Department of Computer Science and Engineering
4-192 EE/CSci Building, 200 Union Street SE, Minneapolis, MN 55455, USA
E-mail: kumar@cs.umn.edu

Youngsong Mun
SoongSil University, School of Computing, Computer Communication Laboratory
1-1 Sang-do 5 Dong, Dong-jak Ku, Seoul 156-743, Korea
E-mail: mun@computing.soongsil.ac.kr

C.J. Kenneth Tan
Queen's University Belfast, Heuchera Technologies Ltd.
Lanyon North, University Road, Belfast, Northern Ireland, BT7 1NN, UK
E-mail: cjtan@optimanumerics.com

Osvaldo Gervasi
University of Perugia, Department of Mathematics and Computer Science
Via Vanvitelli, 1, 06123 Perugia, Italy
E-mail: ogervasi@computer.org

Library of Congress Control Number: 2004105531

CR Subject Classification (1998): D, F, G, H, I, J, C.2-3

ISSN 0302-9743
ISBN 3-540-22057-7 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11010111 06/3142 5 4 3 2 1 0

Preface

The natural mission of Computational Science is to tackle all sorts of human problems and to work out *intelligent* automata aimed at alleviating the burden of working out suitable tools for solving complex problems. For this reason Computational Science, though originating from the need to solve the most challenging problems in science and engineering (computational science is the key player in the fight to gain fundamental advances in astronomy, biology, chemistry, environmental science, physics and several other scientific and engineering disciplines) is increasingly turning its attention to all fields of human activity.

In all activities, in fact, intensive computation, information handling, knowledge synthesis, the use of ad-hoc devices, etc. increasingly need to be exploited and coordinated regardless of the location of both the users and the (various and heterogeneous) computing platforms. As a result the key to understanding the explosive growth of this discipline lies in two adjectives that more and more appropriately refer to Computational Science and its applications: interoperable and ubiquitous. Numerous examples of ubiquitous and interoperable tools and applications are given in the present four LNCS volumes containing the contributions delivered at the 2004 International Conference on Computational Science and its Applications (ICCSA 2004) held in Assisi, Italy, May 14–17, 2004.

To emphasize this particular connotation of modern Computational Science the conference was preceded by a tutorial on Grid Computing (May 13–14) concertedly organized with the COST D23 Action (METACHEM: Metalaboratories for Complex Computational Applications in Chemistry) of the European Coordination Initiative COST in Chemistry and the Project *Enabling Platforms for High-Performance Computational Grids Oriented to Scalable Virtual Organization* of the Ministry of Science and Education of Italy.

The volumes consist of 460 peer reviewed papers given as oral contributions at the conference. The conference included 8 presentations from keynote speakers, 15 workshops and 3 technical sessions. Thanks are due to most of the workshop organizers and the Program Committee members, who took care of the unexpected exceptional load of reviewing work (either carrying it out by themselves or distributing it to experts in the various fields).

Special thanks are due to Noelia Faginas Lago for handling all the necessary secretarial work. Thanks are also due to the young collaborators of the High Performance Computing and the Computational Dynamics and Kinetics research groups of the Department of Mathematics and Computer Science and of the Department of Chemistry of the University of Perugia. Thanks are, obviously,

due as well to the sponsors for supporting the conference with their financial and organizational help.

May 2004

Antonio Laganà
on behalf of the co-editors:
Marina L. Gavrilova
Vipin Kumar
Youngsong Mun
C.J. Kenneth Tan
Osvaldo Gervasi

Organization

ICCSA 2004 was organized by the University of Perugia, Italy; the University of Minnesota, Minneapolis (MN), USA and the University of Calgary, Calgary (Canada).

Conference Chairs

Osvaldo Gervasi (University of Perugia, Perugia, Italy), Conference Chair
Marina L. Gavrilova (University of Calgary, Calgary, Canada),
Conference Co-chair
Vipin Kumar (University of Minnesota, Minneapolis, USA), Honorary Chair

International Steering Committee

J.A. Rod Blais (University of Calgary, Canada)
Alexander V. Bogdanov (Institute for High Performance Computing and Data Bases, Russia)
Marina L. Gavrilova (University of Calgary, Canada)
Andres Iglesias (University de Cantabria, Spain)
Antonio Laganà (University of Perugia, Italy)
Vipin Kumar (University of Minnesota, USA)
Youngsong Mun (Soongsil University, Korea)
Reneé S. Renner (California State University at Chico, USA)
C.J. Kenneth Tan (Heuchera Technologies, Canada and The Queen's University of Belfast, UK)

Local Organizing Committee

Osvaldo Gervasi (University of Perugia, Italy)
Antonio Laganà (University of Perugia, Italy)
Noelia Faginas Lago (University of Perugia, Italy)
Sergio Tasso (University of Perugia, Italy)
Antonio Riganelli (University of Perugia, Italy)
Stefano Crocchianti (University of Perugia, Italy)
Leonardo Pacifici (University of Perugia, Italy)
Cristian Dittamo (University of Perugia, Italy)
Matteo Lobbiani (University of Perugia, Italy)

Workshop Organizers

Information Systems and Information Technologies (ISIT)

Youngsong Mun (Soongsil University, Korea)

Approaches or Methods of Security Engineering

Haeng Kon Kim (Catholic University of Daegu, Daegu, Korea)
Tai-hoon Kim (Korea Information Security Agency, Korea)

Authentication Technology

Eui-Nam Huh (Seoul Women's University, Korea)
Ki-Young Mun (Seoul Women's University, Korea)
Taemyung Chung (Seoul Women's University, Korea)

Internet Communications Security

José Sierra-Camara (ITC Security Lab., University Carlos III of Madrid, Spain)
Julio Hernandez-Castro (ITC Security Lab., University Carlos III of Madrid, Spain)
Antonio Izquierdo (ITC Security Lab., University Carlos III of Madrid, Spain)

Location Management and Security in Next Generation Mobile Networks

Dong Chun Lee (Howon University, Chonbuk, Korea)
Kuinam J. Kim (Kyonggi University, Seoul, Korea)

Routing and Handoff

Hyunseung Choo (Sungkyunkwan University, Korea)
Frederick T. Sheldon (Sungkyunkwan University, Korea)
Alexey S. Rodionov (Sungkyunkwan University, Korea)

Grid Computing

Peter Kacsuk (MTA SZTAKI, Budapest, Hungary)
Robert Lovas (MTA SZTAKI, Budapest, Hungary)

Resource Management and Scheduling Techniques for Cluster and Grid Computing Systems

Jemal Abawajy (Carleton University, Ottawa, Canada)

Parallel and Distributed Computing

Jiawan Zhang (Tianjin University, Tianjin, China)
Qi Zhai (Tianjin University, Tianjin, China)
Wenxuan Fang (Tianjin University, Tianjin, China)

Molecular Processes Simulations

Antonio Laganà (University of Perugia, Perugia, Italy)

Numerical Models in Biomechanics

Jiri Nedoma (Academy of Sciences of the Czech Republic, Prague, Czech Republic)
Josef Danek (University of West Bohemia, Pilsen, Czech Republic)

Scientific Computing Environments (SCEs) for Imaging in Science

Almerico Murli (University of Naples Federico II and Institute for High Performance Computing and Networking, ICAR, Italian National Research Council, Naples, Italy)
Giuliano Laccetti (University of Naples Federico II, Naples, Italy)

Computer Graphics and Geometric Modeling (TSCG 2004)

Andres Iglesias (University of Cantabria, Santander, Spain)
Deok-Soo Kim (Hanyang University, Seoul, Korea)

Virtual Reality in Scientific Applications and Learning

Osvaldo Gervasi (University of Perugia, Perugia, Italy)

Web-Based Learning

Woochun Jun (Seoul National University of Education, Seoul, Korea)

Matrix Approximations with Applications to Science, Engineering and Computer Science

Nicoletta Del Buono (University of Bari, Bari, Italy)
Tiziano Politi (Politecnico di Bari, Bari, Italy)

Spatial Statistics and Geographic Information Systems: Algorithms and Applications

Stefania Bertazzon (University of Calgary, Calgary, Canada)
Borruso Giuseppe (University of Trieste, Trieste, Italy)

Computational Geometry and Applications (CGA 2004)

Marina L. Gavrilova (University of Calgary, Calgary, Canada)

Program Committee

Jemal Abawajy (Carleton University, Canada)
Kenny Adamson (University of Ulster, UK)
Stefania Bertazzon (University of Calgary, Canada)
Sergei Bespamyatnikh (Duke University, USA)
J.A. Rod Blais (University of Calgary, Canada)
Alexander V. Bogdanov (Institute for High Performance Computing and Data Bases, Russia)
Richard P. Brent(Oxford University, UK)
Martin Buecker (Aachen University, Germany)
Rajkumar Buyya (University of Melbourne, Australia)
Hyunseung Choo (Sungkyunkwan University, Korea)
Toni Cortes (Universidad de Catalunya, Barcelona, Spain)
Danny Crookes (The Queen's University of Belfast, (UK))
Brian J. d'Auriol (University of Texas at El Paso, USA)
Ivan Dimov (Bulgarian Academy of Sciences, Bulgaria)
Matthew F. Dixon (Heuchera Technologies, UK)
Marina L. Gavrilova (University of Calgary, Canada)
Osvaldo Gervasi (University of Perugia, Italy)
James Glimm (SUNY Stony Brook, USA)
Christopher Gold (Hong Kong Polytechnic University, Hong Kong, ROC)
Paul Hovland (Argonne National Laboratory, USA)
Andres Iglesias (University de Cantabria, Spain)
Elisabeth Jessup (University of Colorado, USA)
Chris Johnson (University of Utah, USA)
Peter Kacsuk (Hungarian Academy of Science, Hungary)
Deok-Soo Kim (Hanyang University, Korea)
Vipin Kumar (University of Minnesota, USA)
Antonio Laganà (University of Perugia, Italy)
Michael Mascagni (Florida State University, USA)
Graham Megson (University of Reading, UK)
Youngsong Mun (Soongsil University, Korea)
Jiri Nedoma (Academy of Sciences of the Czech Republic, Czech Republic)
Robert Panoff (Shodor Education Foundation, USA)
Reneé S. Renner (California State University at Chico, USA)
Heather J. Ruskin (Dublin City University, Ireland)
Muhammad Sarfraz (King Fahd University of Petroleum and Minerals, Saudi Arabia)
Edward Seidel (Louisiana State University, (USA) and Albert-Einstein-Institut, Potsdam, Germany)
Vaclav Skala (University of West Bohemia, Czech Republic)
Masha Sosonkina (University of Minnesota, (USA))
David Taniar (Monash University, Australia)
Ruppa K. Thulasiram (University of Manitoba, Canada)
Koichi Wada (University of Tsukuba, Japan)

Stephen Wismath (University of Lethbridge, Canada)
Chee Yap (New York University, USA)
Osman Yaşar (SUNY at Brockport, USA)

Sponsoring Organizations

University of Perugia, Perugia, Italy

University of Calgary, Calgary, Canada

University of Minnesota, Minneapolis, MN, USA

The Queen's University of Belfast, UK

Heuchera Technologies, UK

The project **GRID.IT**: *Enabling Platforms for High-Performance Computational Grids Oriented to Scalable Virtual Organizations*, of the Ministry of Science and Education of Italy

COST – European Cooperation in the Field of Scientific and Technical Research

Table of Contents – Part III

Workshop on Computational Geometry and Applications (CGA 04)

Geometric Graphs Realization as Coin Graphs 1
Manuel Abellanas, Carlos Moreno-Jiménez

Disc Covering Problem with Application to Digital Halftoning 11
Tetsuo Asano, Peter Brass, Shinji Sasahara

On Local Transformations in Plane Geometric Graphs Embedded on Small Grids 22
Manuel Abellanas, Prosenjit Bose, Alfredo García, Ferran Hurtado, Pedro Ramos, Eduardo Rivera-Campo, Javier Tejel

Reducing the Time Complexity of Minkowski-Sum Based Similarity Calculations by Using Geometric Inequalities 32
Henk Bekker, Axel Brink

A Practical Algorithm for Approximating Shortest Weighted Path between a Pair of Points on Polyhedral Surface 42
Sasanka Roy, Sandip Das, Subhas C. Nandy

Plane-Sweep Algorithm of O(nlogn) for the Inclusion Hierarchy among Circles 53
Deok-Soo Kim, Byunghoon Lee, Cheol-Hyung Cho, Kokichi Sugihara

Shortest Paths for Disc Obstacles 62
Deok-Soo Kim, Kwangseok Yu, Youngsong Cho, Donguk Kim, Chee Yap

Improving the Global Continuity of the Natural Neighbor Interpolation 71
Hisamoto Hiyoshi, Kokichi Sugihara

Combinatories and Triangulations 81
Tomas Hlavaty, Václav Skala

Approximations for Two Decomposition-Based Geometric Optimization Problems 90
Minghui Jiang, Brendan Mumey, Zhongping Qin, Andrew Tomascak, Binhai Zhu

Computing Largest Empty Slabs 99
Jose Miguel Díaz-Báñez, Mario Alberto López, Joan Antoni Sellarès

3D-Color-Structure-Code – A New Non-plainness Island Hierarchy 109
Patrick Sturm

Quadratic-Time Linear-Space Algorithms for Generating Orthogonal Polygons with a Given Number of Vertices 117
Ana Paula Tomás, António Leslie Bajuelos

Partitioning Orthogonal Polygons by Extension of All Edges Incident to Reflex Vertices: Lower and Upper Bounds on the Number of Pieces 127
António Leslie Bajuelos, Ana Paula Tomás, Fábio Marques

On the Time Complexity of Rectangular Covering Problems in the Discrete Plane 137
Stefan Porschen

Approximating Smallest Enclosing Balls 147
Frank Nielsen, Richard Nock

Geometry Applied to Designing Spatial Structures: Joining Two Worlds 158
José Andrés Díaz, Reinaldo Togores, César Otero

A Robust and Fast Algorithm for Computing Exact and Approximate Shortest Visiting Routes 168
Håkan Jonsson

Automated Model Generation System Based on Freeform Deformation and Genetic Algorithm 178
Hyunpung Park, Kwan H. Lee

Speculative Parallelization of a Randomized Incremental Convex Hull Algorithm 188
Marcelo Cintra, Diego R. Llanos, Belén Palop

The Employment of Regular Triangulation for Constrained Delaunay Triangulation 198
Pavel Maur, Ivana Kolingerová

The Anchored Voronoi Diagram 207
Jose Miguel Díaz-Báñez, Francisco Gómez, Immaculada Ventura

Implementation of the Voronoi-Delaunay Method for Analysis of Intermolecular Voids 217
A.V. Anikeenko, M.G. Alinchenko, V.P. Voloshin, N.N. Medvedev, M.L. Gavrilova, P. Jedlovszky

Approximation of the Boat-Sail Voronoi Diagram and Its Application 227
Tetsushi Nishida, Kokichi Sugihara

Incremental Adaptive Loop Subdivision 237
Hamid-Reza Pakdel, Faramarz F. Samavati

Reverse Subdivision Multiresolution for Polygonal Silhouette Error Correction 247
Kevin Foster, Mario Costa Sousa, Faramarz F. Samavati, Brian Wyvill

Cylindrical Approximation of a Neuron from Reconstructed Polyhedron 257
Wenhao Lin, Binhai Zhu, Gwen Jacobs, Gary Orser

Skeletizing 3D-Objects by Projections 267
David Ménegaux, Dominique Faudot, Hamamache Kheddouci

Track on Computational Geometry

An Efficient Algorithm for Determining 3-D Bi-plane Imaging Geometry 277
Jinhui Xu, Guang Xu, Zhenming Chen, Kenneth R. Hoffmann

Error Concealment Method Using Three-Dimensional Motion Estimation 288
Dong-Hwan Choi, Sang-Hak Lee, Chan-Sik Hwang

Confidence Sets for the Aumann Mean of a Random Closed Set 298
Raffaello Seri, Christine Choirat

An Algorithm of Mapping Additional Scalar Value in 2D Vector Field Visualization 308
Zhigeng Pan, Jianfeng Lu, Minming Zhang

Network Probabilistic Connectivity: Exact Calculation with Use of Chains 315
Olga K. Rodionova, Alexey S. Rodionov, Hyunseung Choo

Curvature Dependent Polygonization by the Edge Spinning 325
Martin Čermák, Václav Skala

SOM: A Novel Model for Defining Topological Line-Region Relations 335
Xiaolin Wang, Yingwei Luo, Zhuoqun Xu

Track on Adaptive Algorithms

On Automatic Global Error Control in Multistep Methods with Polynomial Interpolation of Numerical Solution 345
Gennady Yu. Kulikov, Sergey K. Shindin

Approximation Algorithms for k-Source Bottleneck Routing Cost Spanning Tree Problems 355
Yen Hung Chen, Bang Ye Wu, Chuan Yi Tang

Efficient Sequential and Parallel Algorithms for Popularity Computation on the World Wide Web with Applications against Spamming 367
Sung-Ryul Kim

Decentralized Inter-agent Message Forwarding Protocols for Mobile Agent Systems 376
JinHo Ahn

Optimization of Usability on an Authentication System Built from Voice and Neural Networks 386
Tae-Seung Lee, Byong-Won Hwang

An Efficient Simple Cooling Schedule for Simulated Annealing 396
Mir M. Atiqullah

A Problem-Specific Convergence Bound for Simulated Annealing-Based Local Search 405
Andreas A. Albrecht

Comparison and Selection of Exact and Heuristic Algorithms 415
Joaquín Pérez O., Rodolfo A. Pazos R., Juan Frausto-Solís, Guillermo Rodríguez O., Laura Cruz R., Héctor Fraire H.

Adaptive Texture Recognition in Image Sequences with Prediction through Features Interpolation 425
Sung Baik, Ran Baik

Fuzzy Matching of User Profiles for a Banner Engine 433
Alfredo Milani, Chiara Morici, Radoslaw Niewiadomski

Track on Biology, Biochemistry, Bioinformatics

Genome Database Integration 443
Andrew Robinson, Wenny Rahayu

Protein Structure Prediction with Stochastic Optimization Methods: Folding and Misfolding the Villin Headpiece 454
Thomas Herges, Alexander Schug, Wolfgang Wenzel

High Throughput *in-silico* Screening against Flexible Protein Receptors 465
Holger Merlitz, Wolfgang Wenzel

A Sequence-Focused Parallelisation of EMBOSS on a Cluster of Workstations 473
Karl Podesta, Martin Crane, Heather J. Ruskin

A Parallel Solution to Reverse Engineering Genetic Networks 481
Dorothy Bollman, Edusmildo Orozco, Oscar Moreno

Deformable Templates for Recognizing the Shape of the Zebra Fish Egg Cell 489
Ho-Dong Lee, Min-Soo Jang, Seok-Joo Lee, Yong-Guk Kim, Byungkyu Kim, Gwi-Tae Park

Multiple Parameterisation of Human Immune Response in HIV: Many-Cell Models 498
Yu Feng, Heather J. Ruskin, Yongle Liu

Track on Cluster Computing

Semantic Completeness in Sub-ontology Extraction Using Distributed Methods 508
Mehul Bhatt, Carlo Wouters, Andrew Flahive, Wenny Rahayu, David Taniar

Distributed Mutual Exclusion Algorithms on a Ring of Clusters 518
Kayhan Erciyes

A Cluster Based Hierarchical Routing Protocol for Mobile Networks 528
Kayhan Erciyes, Geoffrey Marshall

Distributed Optimization of Fiber Optic Network Layout Using MATLAB 538
Roman Pfarrhofer, Markus Kelz, Peter Bachhiesl, Herbert Stögner, Andreas Uhl

Cache Conscious Dynamic Transaction Routing in a Shared Disks Cluster 548
Kyungoh Ohn, Haengrae Cho

A Personalized Recommendation Agent System for E-mail Document Classification 558
Ok-Ran Jeong, Dong-Sub Cho

An Adaptive Prefetching Method for Web Caches 566
Jaeeun Jeon, Gunhoon Lee, Ki Dong Lee, Byoungchul Ahn

Track on Computational Medicine

Image Processing and Retinopathy: A Novel Approach to Computer Driven Tracing of Vessel Network 575
Annamaria Zaia, Pierluigi Maponi, Maria Marinelli, Anna Piantanelli, Roberto Giansanti, Roberto Murri

Automatic Extension of Korean Predicate-Based Sub-categorization Dictionary from Sense Tagged Corpora 585
Kyonam Choo, Seokhoon Kang, Hongki Min, Yoseop Woo

Information Fusion for Probabilistic Reasoning and Its Application to the Medical Decision Support Systems 593
Michal Wozniak

Robust Contrast Enhancement for Microcalcification in Mammography 602
Ho-Kyung Kang, Nguyen N. Thanh, Sung-Min Kim, Yong Man Ro

Track on Computational Methods

Exact and Approximate Algorithms for Two–Criteria Topological Design Problem of WAN with Budget and Delay Constraints 611
Mariusz Gola, Andrzej Kasprzak

Data Management with Load Balancing in Distributed Computing 621
Jong Sik Lee

High Performance Modeling with Quantized System 630
Jong Sik Lee

New Digit-Serial Systolic Arrays for Power-Sum and Division Operation in GF(2^m) 638
Won-Ho Lee, Keon-Jik Lee, Kee-Young Yoo

Generation of Unordered Binary Trees 648
Brice Effantin

A New Systolic Array for Least Significant Digit First Multiplication in $GF(2^m)$ 656
Chang Hoon Kim, Soonhak Kwon, Chun Pyo Hong, Hiecheol Kim

Asymptotic Error Estimate of Iterative Newton-Type Methods and Its Practical Application 667
Gennady Yu. Kulikov, Arkadi I. Merkulov

Numerical Solution of Linear High-Index DAEs 676
Mohammad Mahdi Hosseini

Fast Fourier Transform for Option Pricing: Improved Mathematical Modeling and Design of Efficient Parallel Algorithm 686
Sajib Barua, Ruppa K. Thulasiram, Parimala Thulasiraman

Global Concurrency Control Using Message Ordering of Group Communication in Multidatabase Systems 696
Aekyung Moon, Haengrae Cho

Applications of Fuzzy Data Mining Methods for Intrusion DetectionSystems 706
Jian Guan, Da-xin Liu, Tong Wang

Pseudo-Random Binary Sequences Synchronizer Based on Neural Networks 715
Jan Borgosz, Boguslaw Cyganek

Calculation of the Square Matrix Determinant: Computational Aspects and Alternative Algorithms 722
Antonio Annibali, Francesco Bellini

Differential Algebraic Method for Aberration Analysis of Electron Optical Systems 729
Min Cheng, Yilong Lu, Zhenhua Yao

Optimizing Symmetric FFTs with Prime Edge-Length 736
Edusmildo Orozco, Dorothy Bollman

A Spectral Technique to Solve the Chromatic Number Problem in Circulant Graphs 745
Monia Discepoli, Ivan Gerace, Riccardo Mariani, Andrea Remigi

A Method to Establish the Cooling Scheme in Simulated Annealing Like Algorithms 755
Héctor Sanvicente-Sánchez, Juan Frausto-Solís

Packing: Scheduling, Embedding, and Approximating Metrics 764
Hu Zhang

Track on Computational Science Education

Design Patterns in Scientific Software 776
Henry Gardner

Task Modeling in Computer Supported Collaborative Learning Environments to Adapt to Mobile Computing 786
Ana I. Molina, Miguel A. Redondo, Manuel Ortega

Computational Science and Engineering (CSE) Education: Faculty and Student Perspectives 795
Hasan Dağ, Gürkan Soykan, Şenol Pişkin, Osman Yaşar

Computational Math, Science, and Technology: A New Pedagogical Approach to Math and Science Education 807
Osman Yaşar

Track on Computer Modeling and Simulation

Resonant Tunneling Heterostructure Devices – Dependencies on Thickness and Number of Quantum Wells 817
Nenad Radulovic, Morten Willatzen, Roderick V.N. Melnik

Teletraffic Generation of Self-Similar Processes with Arbitrary Marginal Distributions for Simulation: Analysis of Hurst Parameters 827
Hae-Duck J. Jeong, Jong-Suk Ruth Lee, Hyoung-Woo Park

Design, Analysis, and Optimization of LCD Backlight Unit Using Ray Tracing Simulation 837
Joonsoo Choi, Kwang-Soo Hahn, Heekyung Seo, Seong-Cheol Kim

An Efficient Parameter Estimation Technique for a Solute Transport Equation in Porous Media 847
Jaemin Ahn, Chung-Ki Cho, Sungkwon Kang, YongHoon Kwon

HierGen: A Computer Tool for the Generation of Activity-on-the-Node Hierarchical Project Networks 857
Miguel Gutiérrez, Alfonso Durán, David Alegre, Francisco Sastrón

Macroscopic Treatment to Polymorphic E-mail Based Viruses 867
Cholmin Kim, Soung-uck Lee, Manpyo Hong

Making Discrete Games 877
Inmaculada García, Ramón Mollá

Speech Driven Facial Animation Using Chinese Mandarin Pronunciation Rules 886
Mingyu You, Jiajun Bu, Chun Chen, Mingli Song

Autonomic Protection System Using Adaptive Security Policy 896
Sihn-hye Park, Wonil Kim, Dong-kyoo Kim

A Novel Method to Support User's Consent in Usage Control for Stable Trust in E-business 906
Gunhee Lee, Wonil Kim, Dong-kyoo Kim

Track on Financial and Economical Modeling

No Trade under Rational Expectations in Economy (A Multi-modal Logic Approach) 915
Takashi Matsuhisa

A New Approach for Numerical Identification of Optimal Exercise Curve 926
Chung-Ki Cho, Sunbu Kang, Taekkeun Kim, YongHoon Kwon

Forecasting the Volatility of Stock Index Returns: A Stochastic Neural Network Approach 935
Chokri Slim

Track on Mobile Computing Systems

A New IP Paging Protocol for Hierarchical Mobile IPv6 945
Myung-Kyu Yi, Chong-Sun Hwang

Security Enhanced WTLS Handshake Protocol 955
Jin Kwak, Jongsu Han, Soohyun Oh, Dongho Won

An Adaptive Security Model for Heterogeneous Networks Using MAUT and Simple Heuristics 965
Jongwoo Chae, Ghita Kouadri Mostéfaoui, Mokdong Chung

A New Mechanism for SIP over Mobile IPv6 975
Pyung Soo Kim, Myung Eui Lee, Soohong Park, Young Kuen Kim

A Study for Performance Improvement of Smooth Handoff Using Mobility Management for Mobile IP 985
Kyu-Tae Oh, Jung-Sun Kim

A Fault-Tolerant Protocol for Mobile Agent 993
Guiyue Jin, Byoungchul Ahn, Ki Dong Lee

Performance Analysis of Multimedia Data Transmission with PDA over an Infrastructure Network 1002
Hye-Sun Hur, Youn-Sik Hong

A New Synchronization Protocol for Authentication in Wireless LAN Environment 1010
Hea Suk Jo, Hee Yong Youn

A Study on Secure and Efficient Sensor Network Management Scheme Using PTD 1020
Dae-Hee Seo, Im-Yeong Lee

Author Index 1029

Table of Contents – Part I

Information Systems and Information Technologies (ISIT) Workshop, Multimedia Session

Face Detection by Facial Features with Color Images and Face Recognition Using PCA 1
Jin Ok Kim, Sung Jin Seo, Chin Hyun Chung, Jun Hwang, Woongjae Lee

A Shakable Snake for Estimation of Image Contours 9
Jin-Sung Yoon, Joo-Chul Park, Seok-Woo Jang, Gye-Young Kim

A New Recurrent Fuzzy Associative Memory for Recognizing Time-Series Patterns Contained Ambiguity 17
Joongjae Lee, Won Kim, Jeonghee Cha, Gyeyoung Kim, Hyungil Choi

A Novel Approach for Contents-Based E-catalogue Image Retrieval Based on a Differential Color Edge Model 25
Junchul Chun, Goorack Park, Changho An

A Feature-Based Algorithm for Recognizing Gestures on Portable Computers 33
Mi Gyung Cho, Am Sok Oh, Byung Kwan Lee

Fingerprint Matching Based on Linking Information Structure of Minutiae 41
JeongHee Cha, HyoJong Jang, GyeYoung Kim, HyungIl Choi

Video Summarization Using Fuzzy One-Class Support Vector Machine . . . 49
YoungSik Choi, KiJoo Kim

A Transcode and Prefetch Technique of Multimedia Presentations for Mobile Terminals 57
Maria Hong, Euisun Kang, Sungmin Um, Dongho Kim, Younghwan Lim

Information Systems and Information Technologies (ISIT) Workshop, Algorithm Session

A Study on Generating an Efficient Bottom-up Tree Rewrite Machine for JBurg 65
KyungWoo Kang

A Study on Methodology for Enhancing Reliability of Datapath 73
SunWoong Yang, MoonJoon Kim, JaeHeung Park, Hoon Chang

A Useful Method for Multiple Sequence Alignment and Its Implementation 81
Jin Kim, Dong-Hoi Kim, Saangyong Uhmn

A Research on the Stochastic Model for Spoken Language Understanding 89
Yong-Wan Roh, Kwang-Seok Hong, Hyon-Gu Lee

The Association Rule Algorithm with Missing Data in Data Mining 97
Bobby D. Gerardo, Jaewan Lee, Jungsik Lee, Mingi Park, Malrey Lee

Constructing Control Flow Graph for Java by Decoupling Exception Flow from Normal Flow 106
Jang-Wu Jo, Byeong-Mo Chang

On Negation-Based Conscious Agent 114
Kang Soo Tae, Hee Yong Youn, Gyung-Leen Park

A Document Classification Algorithm Using the Fuzzy Set Theory and Hierarchical Structure of Document 122
Seok-Woo Han, Hye-Jue Eun, Yong-Sung Kim, László T. Kóczy

A Supervised Korean Verb Sense Disambiguation Algorithm Based on Decision Lists of Syntactic Features 134
Kweon Yang Kim, Byong Gul Lee, Dong Kwon Hong

Information Systems and Information Technologies (ISIT) Workshop, Security Session

Network Security Management Using ARP Spoofing 142
Kyohyeok Kwon, Seongjin Ahn, Jin Wook Chung

A Secure and Practical CRT-Based RSA to Resist Side Channel Attacks 150
ChangKyun Kim, JaeCheol Ha, Sung-Hyun Kim, Seokyu Kim, Sung-Ming Yen, SangJae Moon

A Digital Watermarking Scheme in JPEG-2000 Using the Properties of Wavelet Coefficient Sign 159
Han-Ki Lee, Geun-Sil Song, Mi-Ae Kim, Kil-Sang Yoo, Won-Hyung Lee

A Security Proxy Based Protocol for Authenticating the Mobile IPv6 Binding Updates 167
Il-Sun You, Kyungsan Cho

A Fuzzy Expert System for Network Forensics 175
Jung-Sun Kim, Minsoo Kim, Bong-Nam Noh

A Design of Preventive Integrated Security Management System Using Security Labels and a Brief Comparison with Existing Models 183
D.S. Kim, T.M. Chung

The Vulnerability Assessment for Active Networks; Model, Policy, Procedures, and Performance Evaluations 191
Young J. Han, Jin S. Yang, Beom H. Chang, Jung C. Na, Tai M. Chung

Authentication of Mobile Node Using AAA in Coexistence of VPN and Mobile IP 199
Miyoung Kim, Misun Kim, Youngsong Mun

Survivality Modeling for Quantitative Security Assessment in Ubiquitous Computing Systems* 207
Changyeol Choi, Sungsoo Kim, We-Duke Cho

New Approach for Secure and Efficient Metering in the Web Advertising 215
Soon Seok Kim, Sung Kwon Kim, Hong Jin Park

MLS/SDM: Multi-level Secure Spatial Data Model 222
Young-Hwan Oh, Hae-Young Bae

Detection Techniques for ELF Executable File Using Assembly Instruction Searching 230
Jun-Hyung Park, Min-soo Kim, Bong-Nam Noh

Secure Communication Scheme Applying MX Resource Record in DNSSEC Domain 238
Hyung-Jin Lim, Hak-Ju Kim, Tae-Kyung Kim, Tai-Myung Chung

Committing Secure Results with Replicated Servers 246
Byoung Joon Min, Sung Ki Kim, Chaetae Im

Applied Research of Active Network to Control Network Traffic in Virtual Battlefield 254
Won Goo Lee, Jae Kwang Lee

Design and Implementation of the HoneyPot System with Focusing on the Session Redirection 262
Miyoung Kim, Misun Kim, Youngsong Mun

Information Systems and Information Technologies (ISIT) Workshop, Network Session

Analysis of Performance for MCVoD System 270
SeokHoon Kang, IkSoo Kim, Yoseop Woo

A QoS Improvement Scheme for Real-Time Traffic Using IPv6 Flow Labels 278
In Hwa Lee, Sung Jo Kim

Energy-Efficient Message Management Algorithms in HMIPv6 286
Sun Ok Yang, SungSuk Kim, Chong-Sun Hwang, SangKeun Lee

A Queue Management Scheme for Alleviating the Impact of Packet Size on the Achieved Throughput 294
Sungkeun Lee, Wongeun Oh, Myunghyun Song, Hyun Yoe, JinGwang Koh, Changryul Jung

PTrace: Pushback/SVM Based ICMP Traceback Mechanism against DDoS Attack 302
Hyung-Woo Lee, Min-Goo Kang, Chang-Won Choi

Traffic Control Scheme of ABR Service Using NLMS in ATM Network ... 310
Kwang-Ok Lee, Sang-Hyun Bae, Jin-Gwang Koh, Chang-Hee Kwon, Chong-Soo Cheung, In-Ho Ra

Information Systems and Information Technologies (ISIT) Workshop, Grid Session

XML-Based Workflow Description Language for Grid Applications 319
Yong-Won Kwon, So-Hyun Ryu, Chang-Sung Jeong, Hyoungwoo Park

Placement Algorithm of Web Server Replicas 328
Seonho Kim, Miyoun Yoon, Yongtae Shin

XML-OGL: UML-Based Graphical Web Query Language for XML Documents 337
Chang Yun Jeong, Yong-Sung Kim, Yan Ha

Layered Web-Caching Technique for VOD Services 345
Iksoo Kim, Yoseop Woo, Hyunchul Kang, Backhyun Kim, Jinsong Ouyang

QoS-Constrained Resource Allocation for a Grid-Based Multiple Source Electrocardiogram Application 352
Dong Su Nam, Chan-Hyun Youn, Bong Hwan Lee, Gari Clifford, Jennifer Healey

Efficient Pre-fetch and Pre-release Based Buffer Cache Management for Web Applications 360
Younghun Ko, Jaehyoun Kim, Hyunseung Choo

A New Architecture Design for Differentiated Resource Sharing on Grid Service 370
Eui-Nam Huh

An Experiment and Design of Web-Based Instruction Model for Collaboration Learning 378
Duckki Kim, Youngsong Mun

Information Systems and Information Technologies (ISIT) Workshop, Mobile Session

Performance Limitation of STBC OFDM-CDMA Systems in Mobile Fading Channels 386
Young-Hwan You, Tae-Won Jang, Min-Goo Kang, Hyung-Woo Lee, Hwa-Seop Lim, Yong-Soo Choi, Hyoung-Kyu Song

PMEPR Reduction Algorithms for STBC-OFDM Signals 394
Hyoung-Kyu Song, Min-Goo Kang, Ou-Seb Lee, Pan-Yuh Joo, We-Duke Cho, Mi-Jeong Kim, Young-Hwan You

An Efficient Image Transmission System Adopting OFDM Based Sequence Reordering Method in Non-flat Fading Channel 402
JaeMin Kwak, HeeGok Kang, SungEon Cho, Hyun Yoe, JinGwang Koh

The Efficient Web-Based Mobile GIS Service System through Reduction of Digital Map 410
Jong-Woo Kim, Seong-Seok Park, Chang-Soo Kim, Yugyung Lee

Reducing Link Loss in Ad Hoc Networks 418
Sangjoon Park, Eunjoo Jeong, Byunggi Kim

A Web Based Model for Analyzing Compliance of Mobile Content 426
Woojin Lee, Yongsun Cho, Kiwon Chong

Delay and Collision Reduction Mechanism for Distributed Fair Scheduling in Wireless LANs 434
Kee-Hyun Choi, Kyung-Soo Jang, Dong-Ryeol Shin

Approaches or Methods of Security Engineering Workshop

Bit-Serial Multipliers for Exponentiation and Division in $GF(2^m)$ Using Irreducible AOP 442
Yong Ho Hwang, Sang Gyoo Sim, Pil Joong Lee

Introduction and Evaluation of Development System Security Process of ISO/IEC TR 15504 451
Eun-ser Lee, Kyung Whan Lee, Tai-hoon Kim, Il-Hong Jung

Design on Mobile Secure Electronic Transaction Protocol with Component Based Development 461
Haeng-Kon Kim, Tai-Hoon Kim

A Distributed Online Certificate Status Protocol Based on GQ Signature Scheme 471
Dae Hyun Yum, Pil Joong Lee

A Design of Configuration Management Practices and CMPET in Common Criteria Based on Software Process Improvement Activity ... 481
Sun-Myung Hwang

The Design and Development for Risk Analysis Automatic Tool 491
Young-Hwan Bang, Yoon-Jung Jung, Injung Kim, Namhoon Lee, Gang-Soo Lee

A Fault-Tolerant Mobile Agent Model in Replicated Secure Services 500
Kyeongmo Park

Computation of Multiplicative Inverses in GF(2^n) Using Palindromic Representation 510
Hyeong Seon Yoo, Dongryeol Lee

A Study on Smart Card Security Evaluation Criteria for Side Channel Attacks 517
HoonJae Lee, ManKi Ahn, SeonGan Lim, SangJae Moon

User Authentication Protocol Based on Human Memorable Password and Using RSA 527
IkSu Park, SeungBae Park, ByeongKyun Oh

Supporting Adaptive Security Levels in Heterogeneous Environments 537
Ghita Kouadri Mostéfaoui, Mansoo Kim, Mokdong Chung

Intrusion Detection Using Noisy Training Data 547
Yongsu Park, Jaeheung Lee, Yookun Cho

A Study on Key Recovery Agent Protection Profile Having Composition Function 557
Dae-Hee Seo, Im-Yeong Lee, Hee-Un Park

Simulation-Based Security Testing for Continuity of Essential Service 567
Hyung-Jong Kim, JoonMo Kim, KangShin Lee, HongSub Lee, TaeHo Cho

NextPDM: Improving Productivity and Enhancing the Reusability with a Customizing Framework Toolkit 577
Ha Jin Hwang, Soung Won Kim

A Framework for Security Assurance in Component Based Development . 587
Hangkon Kim

An Information Engineering Methodology for the Security Strategy Planning 597
Sangkyun Kim, Choon Seong Leem

A Case Study in Applying Common Criteria to Development Process of Virtual Private Network 608
Sang ho Kim, Choon Seong Leem

A Pointer Forwarding Scheme for Fault-Tolerant Location Management in Mobile Networks 617
Ihn-Han Bae, Sun-Jin Oh

Architecture Environments for E-business Agent Based on Security 625
Ho-Jun Shin, Soo-Gi Lee

Authentication Authorization Accounting (AAA) Workshop

Multi-modal Biometrics System Using Face and Signature 635
Dae Jong Lee, Keun Chang Kwak, Jun Oh Min, Myung Geun Chun

Simple and Efficient Group Key Agreement Based on Factoring 645
Junghyun Nam, Seokhyang Cho, Seungjoo Kim, Dongho Won

On Facial Expression Recognition Using the Virtual Image Masking for a Security System 655
Jin Ok Kim, Kyong Sok Seo, Chin Hyun Chung, Jun Hwang, Woongjae Lee

Secure Handoff Based on Dual Session Keys in Mobile IP with AAA 663
Yumi Choi, Hyunseung Choo, Byong-Lyol Lee

Detection and Identification Mechanism against Spoofed Traffic Using Distributed Agents 673
Mihui Kim, Kijoon Chae

DMKB : A Defense Mechanism Knowledge Base 683
Eun-Jung Choi, Hyung-Jong Kim, Myuhng-Joo Kim

A Fine-Grained Taxonomy of Security Vulnerability in Active Network Environments 693
Jin S. Yang, Young J. Han, Dong S. Kim, Beom H. Chang, Tai M. Chung, Jung C. Na

A New Role-Based Authorization Model
in a Corporate Workflow Systems 701
HyungHyo Lee, SeungYong Lee, Bong-Nam Noh

A New Synchronization Protocol for Authentication
in Wireless LAN Environment 711
Hea Suk Jo, Hee Yong Youn

A Robust Image Authentication Method Surviving Acceptable
Modifications 722
Mi-Ae Kim, Geun-Sil Song, Won-Hyung Lee

Practical Digital Signature Generation Using Biometrics 728
Taekyoung Kwon, Jae-il Lee

Performance Improvement in Mobile IPv6 Using AAA
and Fast Handoff 738
Changnam Kim, Young-Sin Kim, Eui-Nam Huh, Youngsong Mun

An Efficient Key Agreement Protocol for Secure Authentication 746
Young-Sin Kim, Eui-Nam Huh, Jun Hwang, Byung-Wook Lee

A Policy-Based Security Management Architecture Using XML
Encryption Mechanism for Improving SNMPv3 755
Choong Seon Hong, Joon Heo

IDentification Key Based AAA Mechanism in Mobile IP Networks 765
Hoseong Jeon, Hyunseung Choo, Jai-Ho Oh

An Integrated XML Security Mechanism for Mobile Grid Application 776
Kiyoung Moon, Namje Park, Jongsu Jang, Sungwon Sohn, Jaecheol Ryou

Development of XKMS-Based Service Component for Using PKI
in XML Web Services Environment 784
Namje Park, Kiyoung Moon, Jongsu Jang, Sungwon Sohn

A Scheme for Improving WEP Key Transmission between APs
in Wireless Environment 792
Chi Hyung In, Choong Seon Hong, Il Gyu Song

Internet Communication Security Workshop

Generic Construction of Certificateless Encryption 802
Dae Hyun Yum, Pil Joong Lee

Security Issues in Network File Systems 812
Antonio Izquierdo, Jose María Sierra, Julio César Hernández, Arturo Ribagorda

A Content-Independent Scalable Encryption Model 821
Stefan Lindskog, Johan Strandbergh, Mikael Hackman, Erland Jonsson

Fair Exchange to Achieve Atomicity in Payments of High Amounts Using Electronic Cash 831
Magdalena Payeras-Capella, Josep Lluís Ferrer-Gomila, Llorenç Huguet-Rotger

N3: A Geometrical Approach for Network Intrusion Detection at the Application Layer 841
Juan M. Estévez-Tapiador, Pedro García-Teodoro, Jesús E. Díaz-Verdejo

Validating the Use of BAN LOGIC 851
José María Sierra, Julio César Hernández, Almudena Alcaide, Joaquín Torres

Use of Spectral Techniques in the Design of Symmetrical Cryptosystems 859
Luis Javier García Villalba

Load Balancing and Survivability for Network Services Based on Intelligent Agents 868
Robson de Oliveira Albuquerque, Rafael T. de Sousa Jr., Tamer Américo da Silva, Ricardo S. Puttini, Clàudia Jacy Barenco Abbas, Luis Javier García Villalba

A Scalable PKI for Secure Routing in the Internet 882
Francesco Palmieri

Cryptanalysis and Improvement of Password Authenticated Key Exchange Scheme between Clients with Different Passwords 895
Jeeyeon Kim, Seungjoo Kim, Jin Kwak, Dongho Won

Timeout Estimation Using a Simulation Model for Non-repudiation Protocols 903
Mildrey Carbonell, Jose A. Onieva, Javier Lopez, Deborah Galpert, Jianying Zhou

DDoS Attack Defense Architecture Using Active Network Technology 915
Choong Seon Hong, Yoshiaki Kasahara, Dea Hwan Lee

A Voting System with Trusted Verifiable Services 924
Macià Mut Puigserver, Josep Lluís Ferrer Gomila, Llorenç Huguet i Rotger

Chaotic Protocols 938
Mohamed Mejri

Security Consequences of Messaging Hubs in Many-to-Many E-procurement Solutions 949
Eva Ponce, Alfonso Durán, Teresa Sánchez

The SAC Test: A New Randomness Test, with Some Applications to PRNG Analysis 960
Julio César Hernandez, José María Sierra, Andre Seznec

A Survey of Web Services Security 968
Carlos Gutiérrez, Eduardo Fernández-Medina, Mario Piattini

Fair Certified E-mail Protocols with Delivery Deadline Agreement 978
Yongsu Park, Yookun Cho

Location Management and the Security in the Next Generation Mobile Networks Workshop

QS-Ware: The Middleware for Providing QoS and Secure Ability to Web Server 988
Seung-won Shin, Kwang-ho Baik, Ki-Young Kim, Jong-Soo Jang

Implementation and Performance Evaluation of High-Performance Intrusion Detection and Response System 998
Hyeong-Ju Kim, Byoung-Koo Kim, Ik-Kyun Kim

Efficient Key Distribution Protocol for Secure Multicast Communication 1007
Bonghan Kim, Hanjin Cho, Jae Kwang Lee

A Bayesian Approach for Estimating Link Travel Time on Urban Arterial Road Network 1017
Taehyung Park, Sangkeon Lee

Perimeter Defence Policy Model of Cascade MPLS VPN Networks 1026
Won Shik Na, Jeom Goo Kim, Intae Ryoo

Design of Authentication and Key Exchange Protocol in Ethernet Passive Optical Networks 1035
Sun-Sik Roh, Su-Hyun Kim, Gwang-Hyun Kim

Detection of Moving Objects Edges to Implement Home Security System in a Wireless Environment 1044
Yonghak Ahn, Kiok Ahn, Oksam Chae

Reduction Method of Threat Phrases by Classifying Assets 1052
Tai-Hoon Kim, Dong Chun Lee

Anomaly Detection Using Sequential Properties of Packets in Mobile Environment 1060
Seong-sik Hong, Hwang-bin Ryou

A Case Study in Applying Common Criteria to Development Process to Improve Security of Software Products 1069
Sang Ho Kim, Choon Seong Leem

A New Recovery Scheme with Reverse Shared Risk Link Group in GMPLS-Based WDM Networks 1078
Hyuncheol Kim, Seongjin Ahn, Daeho Kim, Sunghae Kim, Jin Wook Chung

Real Time Estimation of Bus Arrival Time under Mobile Environment ... 1088
Taehyung Park, Sangkeon Lee, Young-Jun Moon

Call Tracking and Location Updating Using DHS in Mobile Networks ... 1097
Dong Chun Lee

Routing and Handoff Workshop

Improving TCP Performance over Mobile IPv6 1105
Young-Chul Shim, Nam-Chang Kim, Ho-Seok Kang

Design of Mobile Network Route Optimization Based on the Hierarchical Algorithm 1115
Dongkeun Lee, Keecheon Kim, Sunyoung Han

On Algorithms for Minimum-Cost Quickest Paths with Multiple Delay-Bounds 1125
Young-Cheol Bang, Inki Hong, Sungchang Lee, Byungjun Ahn

A Fast Handover Protocol for Mobile IPv6 Using Mobility Prediction Mechanism 1134
Dae Sun Kim, Choong Seon Hong

The Layer 2 Handoff Scheme for Mobile IP over IEEE 802.11 Wireless LAN 1144
Jongjin Park, Youngsong Mun

Session Key Exchange Based on Dynamic Security Association for Mobile IP Fast Handoff 1151
Hyun Gon Kim, Doo Ho Choi

A Modified AODV Protocol with Multi-paths Considering Classes of Services 1159
Min-Su Kim, Ki Jin Kwon, Min Young Chung, Tae-Jin Lee, Jaehyung Park

Author Index 1169

Table of Contents – Part II

Grid Computing Workshop

Advanced Simulation Technique for Modeling Multiphase Fluid Flow in Porous Media 1
Jong G. Kim, Hyoung Woo Park

The P-GRADE Grid Portal 10
Csaba Németh, Gábor Dózsa, Róbert Lovas, Péter Kacsuk

A Smart Agent-Based Grid Computing Platform 20
Kwang-Won Koh, Hie-Cheol Kim, Kyung-Lang Park, Hwang-Jik Lee, Shin-Dug Kim

Publishing and Executing Parallel Legacy Code Using an OGSI Grid Service 30
T. Delaitre, A. Goyeneche, T. Kiss, S.C. Winter

The PROVE Trace Visualisation Tool as a Grid Service 37
Gergely Sipos, Péter Kacsuk

Privacy Protection in Ubiquitous Computing Based on Privacy Label and Information Flow 46
Seong Oun Hwang, Ki Song Yoon

Resource Management and Scheduling Techniques for Cluster and Grid Computing Systems Workshop

Application-Oriented Scheduling in the Knowledge Grid: A Model and Architecture 55
Andrea Pugliese, Domenico Talia

A Monitoring and Prediction Tool for Time-Constraint Grid Application 66
Abdulla Othman, Karim Djemame, Iain Gourlay

Optimal Server Allocation in Reconfigurable Clusters with Multiple Job Types 76
J. Palmer, I. Mitrani

Design and Evaluation of an Agent-Based Communication Model for a Parallel File System 87
María S. Pérez, Alberto Sánchez, Jemal Abawajy, Víctor Robles, José M. Peña

Task Allocation for Minimizing Programs Completion Time in Multicomputer Systems 97
Gamal Attiya, Yskandar Hamam

Fault Detection Service Architecture for Grid Computing Systems 107
J.H. Abawajy

Adaptive Interval-Based Caching Management Scheme for Cluster Video Server 116
Qin Zhang, Hai Jin, Yufu Li, Shengli Li

A Scalable Streaming Proxy Server Based on Cluster Architecture 126
Hai Jin, Jie Chu, Kaiqin Fan, Zhi Dong, Zhiling Yang

The Measurement of an Optimum Load Balancing Algorithm in a Master/Slave Architecture 136
Finbarr O'Loughlin, Desmond Chambers

Data Discovery Mechanism for a Large Peer-to-Peer Based Scientific Data Grid Environment 146
Azizol Abdullah, Mohamed Othman, Md Nasir Sulaiman, Hamidah Ibrahim, Abu Talib Othman

A DAG-Based XCIGS Algorithm for Dependent Tasks in Grid Environments 158
Changqin Huang, Deren Chen, Qinghuai Zeng, Hualiang Hu

Running Data Mining Applications on the Grid: A Bag-of-Tasks Approach 168
Fabrício A.B. da Silva, Sílvia Carvalho, Hermes Senger, Eduardo R. Hruschka, Cléver R.G. de Farias

Parallel and Distributed Computing Workshop

Application of Block Design to a Load Balancing Algorithm on Distributed Networks 178
Yeijin Lee, Okbin Lee, Taehoon Lee, Ilyong Chung

Maintenance Strategy for Efficient Communication at Data Warehouse .. 186
Hyun Chang Lee, Sang Hyun Bae

Conflict Resolution of Data Synchronization in Mobile Environment 196
YoungSeok Lee, YounSoo Kim, Hoon Choi

A Framework for Orthogonal Data and Control Parallelism Exploitation 206
S. Campa, M. Danelutto

Multiplier with Parallel CSA Using CRT's Specific Moduli $(2^k\text{-}1, 2^k, 2^k\text{+}1)$ 216
Wu Woan Kim, Sang-Dong Jang

Unified Development Solution for Cluster and Grid Computing and Its Application in Chemistry 226
Róbert Lovas, Péter Kacsuk, István Lagzi, Tamás Turányi

Remote Visualization Based on Grid Computing 236
Zhigeng Pan, Bailin Yang, Mingmin Zhang, Qizhi Yu, Hai Lin

Avenues for High Performance Computation on a PC 246
Yu-Fai Fung, M. Fikret Ercan, Wai-Leung Cheung, Gujit Singh

A Modified Parallel Computation Model Based on Cluster 252
Xiaotu Li, Jizhou Sun, Jiawan Zhang, Zhaohui Qi, Gang Li

Parallel Testing Method by Partitioning Circuit Based on the Exhaustive Test 262
Wu Woan Kim

A Parallel Volume Splatting Algorithm Based on PC-Clusters 272
Jiawan Zhang, Jizhou Sun, Yi Zhang, Qianqian Han, Zhou Jin

Molecular Processes Simulation Workshop

Three-Center Nuclear Attraction Integrals for Density Functional Theory and Nonlinear Transformations 280
Hassan Safouhi

Parallelization of Reaction Dynamics Codes Using P-GRADE: A Case Study 290
Ákos Bencsura, György Lendvay

Numerical Implementation of Quantum Fluid Dynamics: A Working Example 300
Fabrizio Esposito

Numerical Revelation and Analysis of Critical Ignition Conditions for Branch Chain Reactions by Hamiltonian Systematization Methods of Kinetic Models 313
Gagik A. Martoyan, Levon A. Tavadyan

Computer Simulations in Ion-Atom Collisions 321
S.F.C. O'Rourke, R.T. Pedlow, D.S.F. Crothers

Bond Order Potentials for a priori Simulations of Polyatomic Reactions 328
Ernesto Garcia, Carlos Sánchez, Margarita Albertí, Antonio Laganà

Inorganic Phosphates Investigation by Support Vector Machine 338
Cinzia Pierro, Francesco Capitelli

Characterization of Equilibrium Structure for N_2-N_2 Dimer in 1.2Å≤R≥2.5Å Region Using DFT Method 350
Ajmal H. Hamdani, S. Shahdin

A Time Dependent Study of the Nitrogen Atom Nitrogen Molecule Reaction 357
Antonio Laganà, Leonardo Pacifici, Dimitris Skouteris

From DFT Cluster Calculations to Molecular Dynamics Simulation of N_2 Formation on a Silica Model Surface 366
M. Cacciatore, A. Pieretti, M. Rutigliano, N. Sanna

Molecular Mechanics and Dynamics Calculations to Bridge Molecular Structure Information and Spectroscopic Measurements on Complexes of Aromatic Compounds 374
G. Pietraperzia, R. Chelli, M. Becucci, Antonio Riganelli, Margarita Alberti, Antonio Laganà

Direct Simulation Monte Carlo Modeling of Non Equilibrium Reacting Flows. Issues for the Inclusion into a ab initio Molecular Processes Simulator 383
D. Bruno, M. Capitelli, S. Longo, P. Minelli

Molecular Simulation of Reaction and Adsorption in Nanochemical Devices: Increase of Reaction Conversion by Separation of a Product from the Reaction Mixture 392
William R. Smith, Martin Lísal

Quantum Generalization of Molecular Dynamics Method. Wigner Approach 402
V. Filinov, M. Bonitz, V. Fortov, P. Levashov

$C_6NH_6^+$ Ions as Intermediates in the Reaction between Benzene and N^+ Ions 412
Marco Di Stefano, Marzio Rosi, Antonio Sgamellotti

Towards a Full Dimensional Exact Quantum Calculation of the Li + HF Reactive Cross Section 422
Antonio Laganà, Stefano Crocchianti, Valentina Piermarini

Conformations of 1,2,4,6-Tetrathiepane 432
Issa Yavari, Arash Jabbari, Shahram Moradi

Fine Grain Parallelization of a Discrete Variable Wavepacket Calculation Using ASSIST-CL 437
Stefano Gregori, Sergio Tasso, Antonio Laganà

Numerical Models in Biomechanics Session

On the Solution of Contact Problems with Visco-Plastic Friction in the Bingham Rheology: An Application in Biomechanics 445
Jiří Nedoma

On the Stress-Strain Analysis of the Knee Replacement 456
J. Daněk, F. Denk, I. Hlaváček, Jiří Nedoma, J. Stehlík, P. Vavřík

Musculoskeletal Modeling of Lumbar Spine under Follower Loads 467
Yoon Hyuk Kim, Kyungsoo Kim

Computational Approach to Optimal Transport Network Construction in Biomechanics 476
Natalya Kizilova

Encoding Image Based on Retinal Ganglion Cell 486
Sung-Kwan Je, Eui-Young Cha, Jae-Hyun Cho

Scientific Computing Environments (SCE's) for Imaging in Science Session

A Simple Data Analysis Method for Kinetic Parameters Estimation from Renal Measurements with a Three-Headed SPECT System 495
Eleonora Vanzi, Andreas Robert Formiconi

Integrating Medical Imaging into a Grid Based Computing Infrastructure 505
Paola Bonetto, Mario Guarracino, Fabrizio Inguglia

Integrating Scientific Software Libraries in Problem Solving Environments: A Case Study with ScaLAPACK 515
L. D'Amore, Mario R. Guarracino, G. Laccetti, A. Murli

Parallel/Distributed Film Line Scratch Restoration by Fusion Techniques 525
G. Laccetti, L. Maddalena, A. Petrosino

An Interactive Distributed Environment for Digital Film Restoration 536
F. Collura, A. Machì, F. Nicotra

Computer Graphics and Geometric Modeling Workshop (TSCG 2004)

On Triangulations 544
Ivana Kolingerová

Probability Distribution of Op-Codes in Edgebreaker 554
Deok-Soo Kim, Cheol-Hyung Cho, Youngsong Cho, Chang Wook Kang, Hyun Chan Lee, Joon Young Park

Polyhedron Splitting Algorithm for 3D Layer Generation 564
Jaeho Lee, Joon Young Park, Deok-Soo Kim, Hyun Chan Lee

Synthesis of Mechanical Structures Using a Genetic Algorithm 573
In-Ho Lee, Joo-Heon Cha, Jay-Jung Kim, M.-W. Park

Optimal Direction for Monotone Chain Decomposition 583
Hayong Shin, Deok-Soo Kim

GTVIS: Fast and Efficient Rendering System for Real-Time Terrain Visualization 592
Russel A. Apu, Marina L. Gavrilova

Target Data Projection in Multivariate Visualization – An Application to Mine Planning 603
Leonardo Soto, Ricardo Sánchez, Jorge Amaya

Parametric Freehand Sketches 613
Ferran Naya, Manuel Contero, Nuria Aleixos, Joaquim Jorge

Variable Level of Detail Strips 622
J.F. Ramos, M. Chover

Bézier Solutions of the Wave Equation 631
J.V. Beltran, J. Monterde

Matlab Toolbox for a First Computer Graphics Course for Engineers 641
Akemi Gálvez, A. Iglesias, César Otero, Reinaldo Togores

A Differential Method for Parametric Surface Intersection 651
A. Gálvez, J. Puig-Pey, A. Iglesias

A Comparison Study of Metaheuristic Techniques for Providing QoS to Avatars in DVE Systems 661
P. Morillo, J.M. Orduña, Marcos Fernández, J. Duato

Visualization of Large Terrain Using Non-restricted Quadtree Triangulations 671
Mariano Pérez, Ricardo Olanda, Marcos Fernández

Boundary Filtering in Surface Reconstruction 682
Michal Varnuška, Ivana Kolingerová

Image Coherence Based Adaptive Sampling for Image Synthesis 693
Qing Xu, Roberto Brunelli, Stefano Messelodi, Jiawan Zhang, Mingchu Li

A Comparison of Multiresolution Modelling in Real-Time Terrain Visualisation 703
C. Rebollo, I. Remolar, M. Chover, J.F. Ramos

Photo-realistic 3D Head Modeling Using Multi-view Images 713
Tong-Yee Lee, Ping-Hsien Lin, Tz-Hsien Yang

Texture Mapping on Arbitrary 3D Surfaces 721
Tong-Yee Lee, Shaur-Uei Yan

Segmentation-Based Interpolation of 3D Medical Images 731
Zhigeng Pan, Xuesong Yin, Guohua Wu

A Bandwidth Reduction Scheme for 3D Texture-Based Volume Rendering on Commodity Graphics Hardware 741
Won-Jong Lee, Woo-Chan Park, Jung-Woo Kim, Tack-Don Han, Sung-Bong Yang, Francis Neelamkavil

An Efficient Image-Based 3D Reconstruction Algorithm for Plants 751
Zhigeng Pan, Weixi Hu, Xinyu Guo, Chunjiang Zhao

Where the Truth Lies (in Automatic Theorem Proving in Elementary Geometry) 761
T. Recio, F. Botana

Helical Curves on Surfaces for Computer-Aided Geometric Design and Manufacturing 771
J. Puig-Pey, Akemi Gálvez, A. Iglesias

An Application of Computer Graphics for Landscape Impact Assessment 779
César Otero, Viola Bruschi, Antonio Cendrero, Akemi Gálvez, Miguel Lázaro, Reinaldo Togores

Fast Stereo Matching Using Block Similarity 789
Han-Suh Koo, Chang-Sung Jeong

View Morphing Based on Auto-calibration for Generation of In-between Views 799
Jin-Young Song, Yong-Ho Hwang, Hyun-Ki Hong

Virtual Reality in Scientific Applications and Learning (VRSAL 2004) Workshop

Immersive Displays Based on a Multi-channel PC Clustered System 809
Hunjoo Lee, Kijong Byun

Virtual Reality Technology Applied to Simulate Construction Processes 817
Alcínia Zita Sampaio, Pedro Gameiro Henriques, Pedro Studer

Virtual Reality Applied to Molecular Sciences 827
Osvaldo Gervasi, Antonio Riganelli, Antonio Laganà

Design and Implementation of an Online 3D Game Engine 837
Hunjoo Lee, Taejoon Park

Dynamically Changing Road Networks – Modelling and Visualization in Real Time 843
Christian Mark, Armin Kaußner, Martin Grein, Hartmut Noltemeier

EoL: A Web-Based Distance Assessment System 854
Osvaldo Gervasi, Antonio Laganà

Discovery Knowledge of User Preferences: Ontologies in Fashion Design Recommender Agent System 863
Kyung-Yong Jung, Young-Joo Na, Dong-Hyun Park, Jung-Hyun Lee

When an Ivy League University Puts Its Courses Online, Who's Going to Need a Local University? 873
Matthew C.F. Lau, Rebecca B.N. Tan

Web-Based Learning Session

Threads in an Undergraduate Course: A Java Example Illuminating Different Multithreading Approaches 882
H. Martin Bücker, Bruno Lang, Hans-Joachim Pflug, Andre Vehreschild

A Comparison of Web Searching Strategies According to Cognitive Styles of Elementary Students 892
Hanil Kim, Miso Yun, Pankoo Kim

The Development and Application of a Web-Based Information Communication Ethics Education System 902
Suk-Ki Hong, Woochun Jun

An Interaction Model for Web-Based Learning: Cooperative Project 913
Eunhee Choi, Woochun Jun, Suk-Ki Hong, Young-Cheol Bang

Observing Standards for Web-Based Learning from the Web 922
Luis Anido, Judith Rodríguez, Manuel Caeiro, Juan Santos

Matrix Approximations with Applications to Science, Engineering, and Computer Science Workshop

On Computing the Spectral Decomposition of Symmetric Arrowhead Matrices 932
Fasma Diele, Nicola Mastronardi, Marc Van Barel, Ellen Van Camp

Relevance Feedback for Content-Based Image Retrieval Using Proximal Support Vector Machine 942
YoungSik Choi, JiSung Noh

Orthonormality-Constrained INDSCAL with Nonnegative Saliences 952
Nickolay T. Trendafilov

Optical Flow Estimation via Neural Singular Value Decomposition Learning 961
Simone Fiori, Nicoletta Del Buono, Tiziano Politi

Numerical Methods Based on Gaussian Quadrature and Continuous Runge-Kutta Integration for Optimal Control Problems 971
Fasma Diele, Carmela Marangi, Stefania Ragni

Graph Adjacency Matrix Associated with a Data Partition 979
Giuseppe Acciani, Girolamo Fornarelli, Luciano Liturri

A Continuous Technique for the Weighted Low-Rank Approximation Problem 988
Nicoletta Del Buono, Tiziano Politi

Spatial Statistics and Geographical Information Systems: Algorithms and Applications

A Spatial Multivariate Approach to the Analysis of Accessibility to Health Care Facilities in Canada 998
Stefania Bertazzon

Density Analysis on Large Geographical Databases. Search for an Index of Centrality of Services at Urban Scale 1009
Giuseppe Borruso, Gabriella Schoier

An Exploratory Spatial Data Analysis (ESDA) Toolkit for the Analysis of Activity/Travel Data 1016
Ronald N. Buliung, Pavlos S. Kanaroglou

Using Formal Ontology for Integrated Spatial Data Mining 1026
Sungsoon Hwang

G.I.S. and Fuzzy Sets for the Land Suitability Analysis 1036
Beniamino Murgante, Giuseppe Las Casas

Intelligent Gis and Retail Location Dynamics: A Multi Agent System Integrated with ArcGis 1046
S. Lombardo, M. Petri, D. Zotta

ArcObjects Development in Zone Design Using Visual Basic for Applications 1057
Sergio Palladini

Searching for 2D Spatial Network Holes 1069
Femke Reitsma, Shane Engel

Extension of Geography Markup Language (GML) for Mobile and Location-Based Applications 1079
Young Soo Ahn, Soon-Young Park, Sang Bong Yoo, Hae-Young Bae

A Clustering Method for Large Spatial Databases 1089
Gabriella Schoier, Giuseppe Borruso

GeoSurveillance: Software for Monitoring Change in Geographic Patterns 1096
Peter Rogerson, Ikuho Yamada

From Axial Maps to Mark Point Parameter Analysis (Ma.P.P.A.) – A GIS Implemented Method to Automate Configurational Analysis 1107
V. Cutini, M. Petri, A. Santucci

Computing Foraging Paths for Shore-Birds Using Fractal Dimensions and Pecking Success from Footprint Surveys on Mudflats: An Application for Red-Necked Stints in the Moroshechnaya River Estuary, Kamchatka-Russian Far East 1117
Falk Huettmann

Author Index 1129

Table of Contents – Part IV

Track on Numerical Methods and Algorithms

New Techniques in Designing Finite Difference Domain Decomposition Algorithm for the Heat Equation 1
Weidong Shen, Shulin Yang

A Fast Construction Algorithm for the Incidence Matrices of a Class of Symmetric Balanced Incomplete Block Designs 11
Ju-Hyun Lee, Sungkwon Kang, Hoo-Kyun Choi

ILUTP_Mem: A Space-Efficient Incomplete LU Preconditioner 20
Tzu-Yi Chen

Optimal Gait Control for a Biped Locomotion Using Genetic Algorithm 29
Jin Geol Kim, SangHo Choi, Ki heon Park

A Bayes Algorithm for the Multitask Pattern Recognition Problem – Direct and Decomposed Independent Approaches 39
Edward Puchala

Energy Efficient Routing with Power Management to Increase Network Lifetime in Sensor Networks 46
Hyung-Wook Yoon, Bo-Hyeong Lee, Tae-Jin Lee, Min Young Chung

New Parameter for Balancing Two Independent Measures in Routing Path 56
Moonseong Kim, Young-Cheol Bang, Hyunseung Choo

A Study on Efficient Key Distribution and Renewal in Broadcast Encryption 66
Deok-Gyu Lee, Im-Yeong Lee

Track on Parallel and Distributed Computing

Self-Tuning Mechanism for Genetic Algorithms Parameters, an Application to Data-Object Allocation in the Web 77
Joaquín Pérez, Rodolfo A. Pazos, Juan Frausto, Guillermo Rodríguez, Laura Cruz, Graciela Mora, Héctor Fraire

Digit-Serial AB^2 Systolic Array for Division in GF(2^m) 87
Nam-Yeun Kim, Kee-Young Yoo

Design and Experiment of a Communication-Aware Parallel Quicksort with Weighted Partition of Processors 97
Sangman Moh, Chansu Yu, Dongsoo Han

A Linear Systolic Array for Multiplication in $GF(2^m)$ for High Speed Cryptographic Processors 106
Soonhak Kwon, Chang Hoon Kim, Chun Pyo Hong

Price Driven Market Mechanism for Computational Grid Resource Allocation 117
Chunlin Li, Zhengding Lu, Layuan Li

A Novel LMS Method for Real-Time Network Traffic Prediction 127
Yang Xinyu, Zeng Ming, Zhao Rui, Shi Yi

Dynamic Configuration between Proxy Caches within an Intranet 137
Víctor J. Sosa Sosa, Juan G. González Serna, Xochitl Landa Miguez, Francisco Verduzco Medina, Manuel A. Valdés Marrero

A Market-Based Scheduler for JXTA-Based Peer-to-Peer Computing System 147
Tan Tien Ping, Gian Chand Sodhy, Chan Huah Yong, Fazilah Haron, Rajkumar Buyya

Reducing on the Number of Testing Items in the Branches of Decision Trees 158
Hyontai Sug

CORBA-Based, Multi-threaded Distributed Simulation of Hierarchical DEVS Models: Transforming Model Structure into a Non-hierarchical One 167
Ki-Hyung Kim, Won-Seok Kang

The Effects of Network Topology on Epidemic Algorithms 177
Jesús Acosta-Elías, Ulises Pineda, Jose Martin Luna-Rivera, Enrique Stevens-Navarro, Isaac Campos-Canton, Leandro Navarro-Moldes

A Systematic Database Summary Generation Using the Distributed Query Discovery System 185
Tae W. Ryu, Christoph F. Eick

Parallel Montgomery Multiplication and Squaring over GF(2^m) Based on Cellular Automata 196
Kyo Min Ku, Kyeoung Ju Ha, Wi Hyun Yoo, Kee Young Yoo

A Decision Tree Algorithm for Distributed Data Mining: Towards Network Intrusion Detection 206
Sung Baik, Jerzy Bala

Maximizing Parallelism for Nested Loops with Non-uniform Dependences 213
Sam Jin Jeong

Fair Exchange to Achieve Atomicity in Payments of High Amounts Using Electronic Cash 223
Magdalena Payeras-Capella, Josep Lluís Ferrer-Gomila, Llorenç Huguet-Rotger

Gossip Based Causal Order Broadcast Algorithm 233
ChaYoung Kim, JinHo Ahn, ChongSun Hwang

Track on Signal Processing

Intermediate View Synthesis from Stereoscopic Videoconference Images 243
Chaohui Lu, Ping An, Zhaoyang Zhang

Extract Shape from Clipart Image Using Modified Chain Code – Rectangle Representation 251
Chang-Gyu Choi, Yongseok Chang, Jung-Hyun Cho, Sung-Ho Kim

Control Messaging Channel for Distributed Computer Systems 261
Bogusław Cyganek, Jan Borgosz

Scene-Based Video Watermarking for Broadcasting Systems 271
Uk-Chul Choi, Yoon-Hee Choi, Dae-Chul Kim, Tae-Sun Choi

Distortion-Free of General Information with Edge Enhanced Error Diffusion Halftoning 281
Byong-Won Hwang, Tae-Ha Kang, Tae-Seung Lee

Enhanced Video Coding with Error Resilience Based on Macroblock Data Manipulation 291
Tanzeem Muzaffar, Tae-Sun Choi

Filtering of Colored Noise for Signal Enhancement 301
Myung Eui Lee, Pyung Soo Kim

Model-Based Human Motion Tracking and Behavior Recognition Using Hierarchical Finite State Automata 311
Jihun Park, Sunghun Park, J.K. Aggarwal

Effective Digital Watermarking Algorithm by Contour Detection 321
Won-Hyuck Choi, Hye-jin Shim, Jung-Sun Kim

New Packetization Method for Error Resilient Video Communications ... 329
Kook-yeol Yoo

A Video Mosaicking Technique with Self Scene Segmentation for Video Indexing 338
Yoon-Hee Choi, Yeong Kyeong Seong, Joo-Young Kim, Tae-Sun Choi

Real-Time Video Watermarking for MPEG Streams 348
Kyung-Pyo Kang, Yoon-Hee Choi, Tae-Sun Choi

A TCP-Friendly Congestion Control Scheme Using Hybrid Approach for Reducing Transmission Delay of Real-Time Video Stream 359
Jong-Un Yang, Jeong-Hyun Cho, Sang-Hyun Bae, In-Ho Ra

Object Boundary Edge Selection Using Level-of-Detail Canny Edges 369
Jihun Park, Sunghun Park

Inverse Dithering through IMAP Estimation 379
Monia Discepoli, Ivan Gerace

A Study on Neural Networks Using Taylor Series Expansion of Sigmoid Activation Function 389
Fevzullah Temurtas, Ali Gulbag, Nejat Yumusak

A Study on Neural Networks with Tapped Time Delays: Gas Concentration Estimation 398
Fevzullah Temurtas, Cihat Tasaltin, Hasan Temurtas, Nejat Yumusak, Zafer Ziya Ozturk

Speech Emotion Recognition and Intensity Estimation 406
Mingli Song, Chun Chen, Jiajun Bu, Mingyu You

Speech Hiding Based on Auditory Wavelet 414
Liran Shen, Xueyao Li, Huiqiang Wang, Rubo Zhang

Automatic Selecting Coefficient for Semi-blind Watermarking 421
Sung-kwan Je, Jae-Hyun Cho, Eui-young Cha

Track on Telecommunications

Network Probabilistic Connectivity: Optimal Structures 431
Olga K. Rodionova, Alexey S. Rodionov, Hyunseung Choo

Differentiated Web Service System through Kernel-Level Realtime Scheduling and Load Balancing 441
Myung-Sub Lee, Chang-Hyeon Park, Young-Ho Sohn

Adaptive CBT/Anycast Routing Algorithm for Multimedia Traffic Overload 451
Kwnag-Jae Lee, Won-Hyuck Choi, Jung-Sun Kim

Achieving Fair New Call CAC for Heterogeneous Services in Wireless Networks 460
SungKee Noh, YoungHa Hwang, KiIl Kim, SangHa Kim

Track on Visualization and Virtual and Augmented Reality

Application of MCDF Operations in Digital Terrain Model Processing 471
Zhiqiang Ma, Anthony Watson, Wanwu Guo

Visual Mining of Market Basket Association Rules 479
Kesaraporn Techapichetvanich, Amitava Datta

Visualizing Predictive Models in Decision Tree Generation 489
Sung Baik, Jerzy Bala, Sung Ahn

Track on Software Engineering

A Model for Use Case Priorization Using Criticality Analysis 496
José Daniel García, Jesús Carretero, José María Pérez, Félix García

Using a Goal-Refinement Tree to Obtain and Refine Organizational Requirements 506
Hugo Estrada, Oscar Pastor, Alicia Martínez, Jose Torres-Jimenez

Using C++ Functors with Legacy C Libraries 514
Jan Broeckhove, Kurt Vanmechelen

Debugging of Java Programs Using HDT with Program Slicing 524
Hoon-Joon Kouh, Ki-Tae Kim, Sun-Moon Jo, Weon-Hee Yoo

Frameworks as Web Services 534
Olivia G. Fragoso Diaz, René Santaolaya Salgado, Isaac M. Vásquez Mendez, Manuel A. Valdés Marrero

Exception Rules Mining Based on Negative Association Rules 543
Olena Daly, David Taniar

A Reduced Codification for the Logical Representation of Job Shop Scheduling Problems 553
Juan Frausto-Solis, Marco Antonio Cruz-Chavez

Action Reasoning with Uncertain Resources 563
Alfredo Milani, Valentina Poggioni

Track on Security Engineering

Software Rejuvenation Approach to Security Engineering 574
Khin Mi Mi Aung, Jong Sou Park

A Rollback Recovery Algorithm for Intrusion Tolerant Intrusion Detection System 584
Myung-Kyu Yi, Chong-Sun Hwang

Design and Implementation of High-Performance Intrusion Detection System 594
Byoung-Koo Kim, Ik-Kyun Kim, Ki-Young Kim, Jong-Soo Jang

An Authenticated Key Agreement Protocol Resistant to a Dictionary Attack 603
Eun-Kyung Ryu, Kee-Won Kim, Kee-Young Yoo

A Study on Marking Bit Size for Path Identification Method: Deploying the Pi Filter at the End Host 611
Soon-Dong Kim, Man-Pyo Hong, Dong-Kyoo Kim

Efficient Password-Based Authenticated Key Agreement Protocol 617
Sung-Woon Lee, Woo-Hun Kim, Hyun-Sung Kim, Kee-Young Yoo

A Two-Public Key Scheme Omitting Collision Problem in Digital Signature 627
Sung Keun Song, Hee Yong Youn, Chang Won Park

A Novel Data Encryption and Distribution Approach for High Security and Availability Using LU Decomposition 637
Sung Jin Choi, Hee Yong Youn

An Efficient Conference Key Distribution System Based on Symmetric Balanced Incomplete Block Design 647
Youngjoo Cho, Changkyun Chi, Ilyong Chung

Multiparty Key Agreement Protocol with Cheater Identification Based on Shamir Secret Sharing 655
Kee-Young Yoo, Eun-Kyung Ryu, Jae-Yuel Im

Security of Shen et al.'s Timestamp-Based Password Authentication Scheme 665
Eun-Jun Yoon, Eun-Kyung Ryu, Kee-Young Yoo

ID-Based Authenticated Multiple-Key Agreement Protocol from Pairings 672
Kee-Won Kim, Eun-Kyung Ryu, Kee-Young Yoo

A Fine-Grained Taxonomy of Security Vulnerability in Active Network Environments 681
Jin S. Yang, Young J. Han, Dong S. Kim, Beom H. Chang, Tai M. Chung, Jung C. Na

A Secure and Flexible Multi-signcryption Scheme 689
Seung-Hyun Seo, Sang-Ho Lee

User Authentication Protocol Based on Human Memorable Password and Using RSA 698
IkSu Park, SeungBae Park, ByeongKyun Oh

Effective Packet Marking Approach to Defend against DDoS Attack 708
Heeran Lim, Manpyo Hong

A Relationship between Security Engineering and Security Evaluation . . . 717
Tai-hoon Kim, Haeng-kon Kim

A Relationship of Configuration Management Requirements between KISEC and ISO/IEC 15408 725
Hae-ki Lee, Jae-sun Shim, Seung Lee, Jong-bu Kim

Track on Information Systems and Information Technology

Term-Specific Language Modeling Approach to Text Categorization 735
Seung-Shik Kang

Context-Based Proofreading of Structured Documents 743
Won-Sung Sohn, Teuk-Seob Song, Jae-Kyung Kim, Yoon-Chul Choy, Kyong-Ho Lee, Sung-Bong Yang, Francis Neelamkavil

Implementation of New CTI Service Platform Using Voice XML 754
Jeong-Hoon Shin, Kwang-Seok Hong, Sung-Kyun Eom

Storing Together the Structural Information of XML Documents in Relational Databases 763
Min Jin, Byung-Joo Shin

Annotation Repositioning Methods in the XML Documents: Context-Based Approach 772
Won-Sung Sohn, Myeong-Cheol Ko, Hak-Keun Kim, Soon-Bum Lim, Yoon-Chul Choy

Isolating and Specifying the Relevant Information of an Organizational Model: A Process Oriented Towards Information System Generation 783
Alicia Martínez, Oscar Pastor, Hugo Estrada

A Weighted Fuzzy Min-Max Neural Network for Pattern Classification and Feature Extraction 791
Ho J. Kim, Tae W. Ryu, Thai T. Nguyen, Joon S. Lim, Sudhir Gupta

The eSAIDA Stream Authentication Scheme 799
Yongsu Park, Yookun Cho

An Object-Oriented Metric to Measure the Degree of Dependency Due to Unused Interfaces 808
René Santaolaya Salgado, Olivia G. Fragoso Diaz, Manuel A. Valdés Marrero, Isaac M. Vásquez Mendez, Sheila L. Delfín Lara

End-to-End QoS Management for VoIP Using DiffServ 818
Eun-Ju Ha, Byeong-Soo Yun

Multi-modal Biometrics System Using Face and Signature 828
Dae Jong Lee, Keun Chang Kwak, Jun Oh Min, Myung Geun Chun

Track on Information Retrieval

Using 3D Spatial Relationships for Image Retrieval by XML Annotation 838
SooCheol Lee, EenJun Hwang, YangKyoo Lee

Association Inlining for Mapping XML DTDs to Relational Tables 849
Byung-Joo Shin, Min Jin

XCRAB: A Content and Annotation-Based Multimedia Indexing and Retrieval System 859
SeungMin Rho, SooCheol Lee, EenJun Hwang, YangKyoo Lee

An Efficient Cache Conscious Multi-dimensional Index Structure 869
Jeong Min Shim, Seok Il Song, Young Soo Min, Jae Soo Yoo

Track on Image Processing

Tracking of Moving Objects Using Morphological Segmentation, Statistical Moments, and Radon Transform 877
Muhammad Bilal Ahmad, Min Hyuk Chang, Seung Jin Park, Jong An Park, Tae Sun Choi

Feature Extraction and Correlation for Time-to-Impact Segmentation Using Log-Polar Images 887
Fernando Pardo, Jose A. Boluda, Esther De Ves

Object Mark Segmentation Algorithm Using Dynamic Programming for Poor Quality Images in Automated Inspection Process 896
Dong-Joong Kang, Jong-Eun Ha, In-Mo Ahn

A Line-Based Pose Estimation Algorithm for 3-D Polyhedral Object Recognition 906
Tae-Jung Lho, Dong-Joong Kang, Jong-Eun Ha

Initialization Method for the Self-Calibration
Using Minimal Two Images 915
Jong-Eun Ha, Dong-Joong Kang

Face Recognition for Expressive Face Images 924
Hyoun-Joo Go, Keun Chang Kwak, Sung-Suk Kim, Myung-Geun Chun

Kolmogorov-Smirnov Test for Image Comparison 933
Eugene Demidenko

Modified Radius-Vector Function for Shape Contour Description 940
Sung Kwan Kang, Muhammad Bilal Ahmad, Jong Hun Chun, Pan Koo Kim, Jong An Park

Image Corner Detection Using Radon Transform 948
Seung Jin Park, Muhammad Bilal Ahmad, Rhee Seung-Hak, Seung Jo Han, Jong An Park

Analytical Comparison of Conventional and MCDF Operations
in Image Processing 956
Yinghua Lu, Wanwu Guo

On Extraction of Facial Features from Color Images 964
Jin Ok Kim, Jin Soo Kim, Young Ro Seo, Bum Ro Lee, Chin Hyun Chung, Key Seo Lee, Wha Young Yim, Sang Hyo Lee

Track on Networking

An Architecture for Mobility Management
in Mobile Computing Networks 974
Dohyeon Kim, Beongku An

An Adaptive Security Model for Heterogeneous Networks
Using MAUT and Simple Heuristics 983
Jongwoo Chae, Ghita Kouadri Mostéfaoui, Mokdong Chung

A Hybrid Restoration Scheme Based on Threshold Reaction Time
in Optical Burst-Switched Networks 994
Hae-Joung Lee, Kyu-Yeop Song, Won-Ho So, Jing Zhang, Debasish Datta, Biswanath Mukherjee, Young-Chon Kim

Author Index 1005

Geometric Graphs Realization as Coin Graphs

Manuel Abellanas and Carlos Moreno-Jiménez

Dpto. Mat. Aplic., FI. UPM., Boadilla del Monte, 28660 Madrid, Spain
mabellanas@fi.upm.es, cmoreno11@hotmail.com

Abstract. Koebe's Theorem [8] proves that any planar graph is the contact graph of a set of coins in the plane. But not any planar geometric graph can be realized as a coin graph (with coins centered at the vertices of the graph). This paper presents an algorithm to decide whether a planar connected geometric graph is a coin graph and to obtain, in the affirmative case, all the coin sets whose contact graphs are the given graph. This result is generalized to other metrics different from the Euclidean metric and is applied to a problem in mechanical gear systems. Two related optimization problems are also considered. They are motivated by graph drawing problems in Geographical Information Systems and Architectural Design Systems.

1 Introduction

Let C be a set of circles in the plane. The *intersection graph* of C is the graph whose vertex set is C and two vertices are adjacent if the corresponding circles intersect. If circles in C do not overlap, intersections reduce to a point, and just occur when they are tangent. If this is the case, the intersection graph of C is called a *contact graph* or a *coin graph*. A *geometric graph* is a graph whose vertices are points and edges are straight line segments.

Koebe's theorem [8] proves that every planar graph can be realized as a coin graph. Nevertheless, not every planar geometric graph can be realized as a coin graph with circles centered in its vertices (see Fig. 2). In this paper we give an algorithm to decide whether a planar connected geometric graph can be realized as a coin graph. In the affirmative case, the algorithm gives all the possible solutions. Figure 1 shows an example of the affirmative case and one of its solutions.

We begin with the problem for the case of trees. If a tree is realizable as a coin graph with coins centered in its vertices, there are usually infinitely many solutions (sets of circles) all of them depending on a parameter. Note that when the radius of one of the circles is fixed, all the other radii are also fixed because of the tangency conditions.

The general (connected) case, in which there can be cycles in the graph, can be solved by computing a spanning tree of the graph and applying to it a small variant of the algorithm for trees. Nevertheless, the existence of a solution when having cycles can be considered as a degenerate case, because input data have to fit some algebraic conditions.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 1–10, 2004.
© Springer-Verlag Berlin Heidelberg 2004

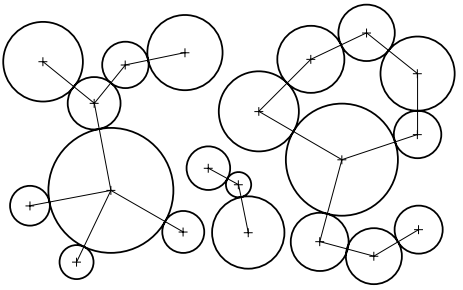


Fig. 1. Geometric graph realizable as a coin graph and one of its solutions

If the graph is not connected, it is possible that while every connected component is realizable, the graph is not, because circles corresponding to different components always overlap.

As we will see, in the efficient solution of the problem, as well as for the related problems, Voronoi diagrams play an important roll; in particular additively weighted Voronoi diagrams. Good references about Voronoi diagrams are [7] and [1]. In [6] one can see a Java applet for computing additively weighted Voronoi diagrams.

In section 2 we give an $O(n \log n)$ algorithm that solves the problem for trees and obtains all possible solutions. In section 3 general connected graphs are considered. Section 4 shows an application to a problem on gear systems in Mechanical Engineering. In section 5, generalizations to other metrics different from the Euclidean metric are considered. Section 6 deals with some optimization related problems and in section 7 several open problems are mentioned.

Results on this paper form part of [5].

2 Solution for Trees

In this section the particular case of trees is considered. The problem is the following:

Problem*: Let T be a geometric planar tree. Verify whether T is realizable as a coin graph with coins centered at the vertices of T. In the affirmative case, obtain all possible solutions.*

Figure 2 shows one of the simplest cases without solution.

The algorithm has three main steps which are described in the following subsections.

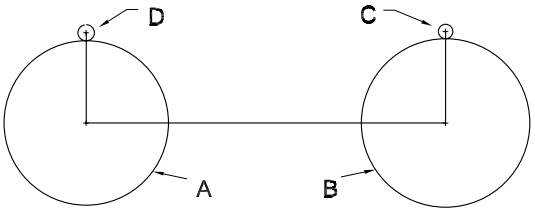


Fig. 2. A non realizable graph. Circles A and B cannot touch due to circles C and D

2.1 Step 1: Tangency Realization

In this step one checks whether it is possible to realize the tree as a coin graph without taking into account possible overlapping between non adjacent circles.

Definition: Let e and f be two edges incident on a common vertex v. Let $\overline{ab}$ be a segment with endpoints a and b contained in e. *To propagate* the segment $\overline{ab}$ from e to f means to obtain the segment contained in f which is the intersection between the edge f and the circular annulus centered in v with radii $dist(v, a)$ and $dist(v, b)$.

Definition: For a given polygonal chain C and a segment $\overline{ab}$ contained in the first edge of the chain, *to propagate the segment* $\overline{ab}$ *along the chain* C means to obtain the segment contained in the last edge of C which is the result of iteratively propagate $\overline{ab}$ from the first edge to the second, the resulting segment to the third one, and so on until the last edge.

Let T be a planar geometric tree. Let us call one of the leaves r of T the *root* of T and let us call the *root edge* the edge adjacent to r. For every leaf l_i of the tree, let S_i be the propagated segment of the edge adjacent to l_i along the path connecting l_i to r in the tree. Let $S = \bigcap_i S_i$. S is a segment contained in the root edge, possibly empty.

Lemma 1. *Every point x interior to S determines a set of circles, $C(x)$, each one of which is centered in one vertex of T and such that two circles centered in adjacent vertices are tangent.*

Proof. It suffices to observe that the propagation of point x along the tree (from the root edge to any other edge in the tree) gives a point in each edge which determines circles centered at the vertices that pass through those points being those points precisely the tangent points between adjacent circles.

Segment S will be called the *propagated segment of the tree*. Endpoints of S, m_1 and m_2, correspond to the cases in which one of the circles shrinks to a point. These correspond to the solutions in which some circles have maximum

size while the others have the minimum size and viceversa. Let us call odd circles the circles centered at vertices with odd distance in the graph to the root and even circles the rest. If m_1 is the endpoint nearest to the root r, note that when point x moves from m_1 to m_2, odd circles grow while even circles decrease.

It is easy to compute the propagated segment of a given tree in linear time.

2.2 Step 2: Circle Inclusion Test

If the propagated segment of the tree is empty, there is no solution. Otherwise it is possible that, for some point $x \in S$, one of the circles in $C(x)$ is contained in the interior of another one (these two circles must correspond to two non adjacent vertices v_i, v_j). In such a case, there is no solution as well, because for every point $x \in S$ the corresponding circles to v_i and v_j overlap.

In order to check this possibility, it suffices to look for the Euclidean nearest vertex of each vertex and check if the Euclidean distance between them is bigger than the difference of the radii of the corresponding circles in $C(m_1)$ as well as in $C(m_2)$.

This checking can be done in $O(n \log n)$ time which is the time needed to compute all nearest neighbors of the vertices.

2.3 Step 3: Non Adjacent Circles Overlapping Test

If the propagated segment of the tree is not empty and no circle in $C(m_1)$ nor in $C(m_2)$ contains another one in its interior, it is possible that two non adjacent circles in $C(m_1)$ or in $C(m_2)$ overlap.

If the path connecting two non adjacent vertices v_i and v_j in the tree has an odd number of edges and the corresponding circles in $C(x)$ intersect, for any $x \in S$, the problem has no solution, because even when the point x moves in S, one of the circles grows while the other shrinks, and therefore they always intersect.

Let us suppose that all overlapping circles in $C(x)$, for all $x \in S$, correspond to vertices connected in the tree with a path with an even number of edges. We measure the overlapping of two circles with the difference between the sum of their radii and the distance between their vertices. Let v_i and v_j be the vertices that correspond to the two circles in $C(m_1)$ with the biggest overlapping. Let r_i and r_j be the corresponding radii. Let $d = r_i + r_j - dist(v_i, v_j)$. If $d/2 < \|S\|$, where $\|S\|$ is the length of S, by reducing S from m_1 an amount $d/2$, one obtains a new segment S with endpoints m_1' and m_2 such that, in $C(m_1')$, no circles overlap. In a similar way, S has to be reduced from m_2 when some circles in $C(m_2)$ overlap.

Lemma 2. *The resulting segment gives all the solutions to the problem.*

Proof. If it is empty, there is no solution. By construction, every point x interior to the segment corresponds to a set of circles $C(x)$ whose intersection graph is the given tree T.

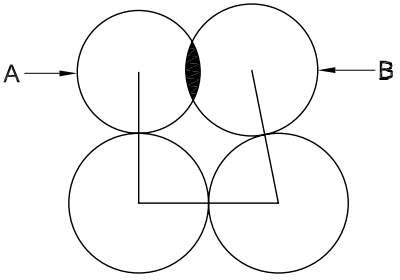


Fig. 3. A non realizable tree (when A shrinks, B grows and viceversa)

A brute force approach for the overlapping test checks every pair of vertices in the tree and reduces the propagated segment or detecting a no solution case. This approach takes $O(n^2)$ time since there are $\binom{n}{2}$ pairs of vertices.

Let us see how by using Voronoi diagrams, it is possible to reduce the time complexity:

Lemma 3. *Detecting the case in which two circles at an odd distance overlap can be done in $O(n \log n)$ time.*

Proof. Consider all circles in $C(m_1)$ whose centers have an odd distance to the root of the tree, and compute their Voronoi diagram. This diagram is the additively weighted Voronoi diagram where sites corresponds to the centers and weights correspond to the radii of the circles (See [7], pg. 133). For every vertex within an even distance to the root in the tree, including the root, locate the Voronoi region containing it and check if the corresponding circles overlap. Proceed in a similar way by changing the rolls of odd and even vertices.

Additively weighted Voronoi diagrams can be computed in $O(n \log n)$ time [7]. Location of each point takes $O(\log n)$ time and the overlapping test for each pair can be done in constant time. Therefore, overall the process takes $O(n \log n)$ time.

Once this process is done, and if there are no overlapping circles at an odd distance, one proceeds with the segment reduction step.

For doing that, one computes the Voronoi diagram of circles in $C(m_1)$ and checks every pair of neighbor circles looking for the pair of circles with biggest overlapping. These two circles determine how much the segment S has to be reduced by one of its end points. A similar procedure with $C(m_2)$ gives the necessary reduction from the other end point.

The time complexity of this step is again $O(n \log n)$, dominated by the construction of Voronoi diagrams.

Finally, one verifies the following theorem:

Theorem 1. *Given a planar geometric tree T, it is possible to decide if it is realizable as a coin graph with coins centered at the vertices of T in $O(n \log n)$*

time. Furthermore, it is possible to describe all the sets of circles centered at the vertices of T whose contact graph is T within the same time bound.

3 Solution for Connected Graphs

For general connected graphs, the idea is to apply the described algorithm to a spanning tree of the graph. A slight modification to the algorithm must be done. In fact, if there is a cycle in the graph, by deleting an edge of the cycle, one obtains a tree. We need a solution for that tree in which the two corresponding circles to the endpoints of the deleted edge are tangent. But actually this would not be a solution for the tree (because these two circles intersect). So step three of the algorithm needs to be modified in order to accept, and verify, tangency between circles that being not adjacent in the tree, are adjacent in the graph.

If there is a cycle in the graph with an even number of edges, there can be an infinite number of solutions, but an algebraic condition must be fulfilled by its vertices. If there is a cycle with an odd number of edges in the graph, then there is at most one solution because circles centered at the endpoints of the deleted edge of the cycle grow or shrink at the same time when varying the radii of circles in the set of solutions for the tree. That means that the existence of an odd cycle fixes the radii of the circles. Therefore, if there are two or more odd cycles, all of them have to give rise to the same solution. If this is the case, there are again algebraic conditions to be fulfilled by the input data. As a consequence, in the general case, the existence of cycles in the graph implies the non existence of solution.

Note that the modification of the algorithm does not affect the overall time complexity which is still $O(n \log n)$.

4 Application to Gear Systems

Previous results can be applied to solve the following problem related to mechanical gear systems:

Problem: *A set of points in the plane represents the axles of a gear system. A geometric connected planar graph, whose vertices are those points, shows the way the gears have to be in contact with each other. The problem is to decide if it is possible to realize a gear system following the design given by the graph, and to obtain all possible solutions.*

A gear system must not be blocked. When one of the gears rotate, all gears also rotate because rotation is transmitted from one gear to all its neighbors and the graph is connected. To not be blocked, a necessary condition is not to have odd length cycles (See Fig. 4).

For solving the problem, it suffices to check this condition and to apply the described algorithm that realizes a geometric graph as a coin graph. The existence of odd cycles in the graph can be checked in linear time. Therefore,

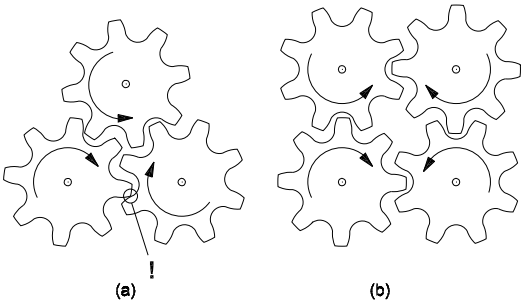


Fig. 4. An odd length cycle blocks the system (a). This is not the case if the cycle has even length (b)

the problem can be solved in $O(n \log n)$ time, where n is the number of gears of the system. As one has seen, the algorithm gives all possible solutions which, in general, depend on one parameter.

In [4] one can see a Java applet that solves the problem obtaining all possible solutions.

5 Generalization to Other Metrics

Coin graphs are contact graphs for discs. That means contact graphs of balls in the Euclidean plane. One can generalize to other metrics, thus obtaining contact graphs of other different shapes. For instance, if one considers L_∞ metric instead of L_2, the problem can be stated as follows:

***Problem**: Let G be a geometric planar graph. Decide if G can be realized as the contact graph of a set of non overlapping isothetic squares centered at the vertices of G and obtain all possible sets of squares whose contact graph is G.*

Figure 5 shows an example with L_1 metric. These two cases are basically the same because they differ by a 45 degree rotation. The more general case in which balls are homothetic rectangles has interest because its applications in architectural design.

In all cases for which Voronoi diagrams (usual and additively weighted) have linear size and can be computed in $O(n \log n)$ time, the proposed algorithm solves the problem in $O(n \log n)$ time as well. Good references for generalizations of Voronoi diagrams are [2] and [7].

6 Optimization Problems

In this section two optimization problems are presented. Note that their solutions do not depend on the metric and they apply whenever the corresponding Voronoi diagrams can be computed in $O(n \log n)$ time.

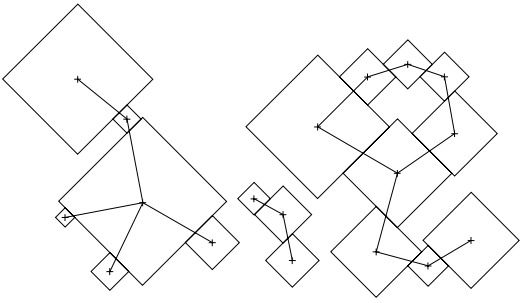


Fig. 5. An example in L_1 metric

6.1 Area Sum Maximization

***Problem**: Given a planar connected geometric graph G, which is realizable as a contact graph of circles centered on its vertices, compute the set of circles whose contact graph is G that maximizes the sum of their areas.*

The algorithm in section 3 and the following lemma are the key for solving this problem.

Lemma 4. *Let G be a graph satisfying the problem conditions and let S be the corresponding segment of solutions obtained by the algorithm in section 3. The sum of the areas of the sets of circles $C(x)$, $x \in S$, is reached in one of the endpoints of S.*

Proof. For every $x \in S$, let $C_i(x)$ be the disc that corresponds to vertex i and let $f_i(x)$ be its area. As function $f_i(x)$ is a quadratic positive function of the radius of $C_i(x)$ and this is a parameter that differs on a constant of $dist(x, m_1)$, being m_1 one of the endpoints of S, one verifies that $\sum_i f_i(x)$ is a quadratic positive function in the variable $dist(x, m_1)$. Therefore it reaches its maximum in one of the extreme points of it domain, which correspond to the endpoints of S.

To solve the problem it suffices to compute the segment of solutions given by the algorithm in section 3 and evaluate the sum of the areas of the set of circles corresponding to the endpoints. The maximum of the two values is the global maximum. All this process can be done in $O(n \log n)$ time in the worst case, because this is the time to compute the segment of solutions, being the rest of the process linear in time.

6.2 Minimum Circle Maximization

***Problem**: Given a planar connected geometric graph G, which is realizable as a contact graph of circles centered on its vertices, compute the set of circles whose contact graph is G and in which the minimum circle is maximized.*

This problem is analogous to the maximization of the lower envelope of the set of functions $f_i(x)$ defined in the previous subsection. Nevertheless, because the perimeter as well as the area of the discs grows with the radius, and radius is a linear function of the value $x \in S$, one can consider the functions which give the radius for each disc, $r_i(x)$, and maximize the lower envelope for these functions. Because they are linear functions, this is a linear programming problem. If Meggido's technique is applied [3], one can obtain the maximum in linear time.

As a consequence, the entire problem can be solved in $O(n \log n)$ time in two steps:

1. Compute the segment of solutions for the realization problem.
2. Maximize the lower envelope of functions $r_i(x)$ with Meggido's algorithm.

6.3 Applications to Graph Drawing

These optimization problems can be applied in the following graph drawing problem. One way of representing a planar graph is by means of non overlapping touching isothetic rectangles. Rectangles are the vertices and two of them are adjacent if they touch each other. Rectangles can be used for labelling the vertices. To do this, it is convenient not to have very small rectangles, because they cannot contain the text of the label properly. One way to avoid this is making a zoom. Another way is to place the rectangles (vertices) in a different position that allows bigger rectangles. Nevertheless, in some applications these two possibilities are not permitted because the geometric location of the vertices is a crucial information of the graph (for instance in Geographic Information Systems or in Architectural Design Systems). It is clear, as we have seen, that a graph cannot always be realized in this way. In the cases in which it is possible, the minimum circle maximization gives the better solution.

7 Open Problems

The geometric graph realization as a coin graph in the non connected case is still unsolved. If the number of connected components of the graph is k, the set of solutions depends in general on k variables. That means that the set of solutions can be seen as a set in $\mathbb{R}^k$. It is an interesting question to study the properties of these kind of sets in $\mathbb{R}^k$.

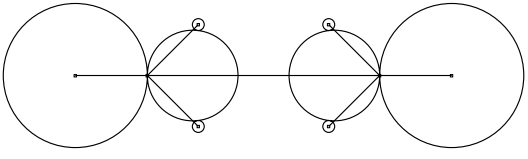


Fig. 6. A non realizable graph even if coins are not centered at the vertices of the graph

As we have seen, not every geometric planar graph is realizable as a coin graph with coins centered at the vertices. This occurs even if the centered condition is relaxed by allowing the coins not to be centered but containing the corresponding vertex inside the coin. Figure 6 shows an example. The realization problem in this case is ongoing work.

Acknowledgments. Thanks are due to Belén Palop and anonymous referees for their interesting comments that allow us to improve the paper. Partially supported by *MCYT* TIC2003-08933-C02-01

References

1. F. Aurenhammer, R. Klein, *Voronoi diagrams*, in Handbook of Computational Geometry, J.R. Sack and J. Urrutia Eds. 201–290, North-Holland (2000).
2. R. Klein, *Concrete and abstract Voronoi diagrams*, LNCS n 400, Springer Verlag (1989).
3. N. Meggido, *Linear Programming in Linear Time when the Dimension is Fixed*, J. Assoc. Comput. Mach. (USA), 31(1), 114–127 (1984).
4. C. Moreno, *An applet to realize a geometric planar graph as a gear system*, *www.dma.fi.upm.es/research/geocomp/coin/circulos.html*
5. C. Moreno, *Algunos problemas de Geometría Computacional en Ingeniería Mecánica*, Ph.D. in preparation.
6. O. Münch, *VoroCircles: an applet to draw Voronoi diagrams with additive weights*, *http://web.informatik.uni-bonn.de/I/GeomLab/apps/voroadd/index.html/* (2001).
7. A. Okabe et al, *Spatial Tessellations: concepts and applications of Voronoi diagrams*, John Wiley & sons, Chichester. (1992).
8. A proof of Koebe's theorem can be found in *Combinatorial Geometry*, by Janos Pach and Pankaj K. Agarwal, John Wiley and Sons. (1995).

Disc Covering Problem with Application to Digital Halftoning

Tetsuo Asano[1], Peter Brass[2], and Shinji Sasahara[1,3]

[1] School of Information Science, JAIST, 1-1 Asahidai, Tatsunokuchi, Ishikawa, 923-1292 Japan. {t-asano,s-sasaha}@jaist.ac.jp
[2] Department of Computer Science, City College, CUNY, Convent Avenue at 138th Street, New York, NY-10031,USA. peter@cs.ccny.cuny.edu
[3] Fuji Xerox Co., Ltd., 430 Sakai, Nakai, Ashigarakami, Kanagawa 259-0157, Japan.

1 Introduction

One of the popular geometric optimization problems is that of finding the maximum radius r_n of n equal and non-overlapping discs to be packed in a unit square. It has been widely explored with a number of surprising results (see e.g. [2]).

The problem to be considered in this paper is similar to the above-stated problems but different in many ways. In our case possible locations of disc centers are restricted to predefined lattice points. Furthermore, the radii of discs are given as a matrix. The problem is to choose discs so that the total area covered by exactly one disc is maximized.

This problem seems to be computationally hard although no proof of its NP-hardness is known. In this paper we first consider the one-dimensional version of the problem and give a polynomial-time algorithm. Then, we propose some approximation algorithms with theoretical guarantee on their performance ratios.

This problem originates from an application to digital halftoning, a technique to convert continuous-tone images into binary images for printers. Our goal for the application to digital halftoning is to distribute points so that the Voronoi diagram associated with them contains no skinny Voronoi region. In this sence the problem is similar to mesh generation, in which the interior of a given polygonal region is partitioned into simplicies to avoid skinny simplices. For the purpose some part may be partitioned into small regions. In our case no polygon is given and the sizes or areas of simplices (convex polygons) are determined by spatial frequency of an image. The idea of using Voronoi diagram for designing screen elements is not new. In fact, it is seen in the literatures [3] and [4], in which Voronoi diagram is used to generate non-regular arrangement of screen elements. The first and third authors [5] formulated the problem as a disc covering problem based on spatial frequency information of an image and presented a heuristic algorithm based on bubble-packing algorithm. It is an iterative improvement algorithm and took much time before conversion. This paper achieves efficient implementation while keeping the quality of the resulting Voronoi diagram and output images.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 11–21, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Motivation and Application

Digital halftoning is a technique to convert a continuous-tone image into a bi-level image for output on bi-level printing devices. Conventional halftoning algorithms are classified into two categories depending on resolution of printing devices. In a low-resolution printer such as an ink-jet printer individual dots are rather clearly separated. On the other hand dots are too small in a high-resolution printer such as off-set printer to make fine control over their positions. Therefore, dots should form clusters whose sizes are determined by their corresponding intensity levels. Such a halftoning algorithm is called a cluster-dot halftoning.

This algorithm consists in partitioning the output image plane into repetitive polygons called screen elements, usually of the same shape such as rectangles or parallelograms. Each screen element is then filled in by dots according to the corresponding intensity levels. Dots in a screen element is clustered around some center point to form a rounded figure. Denoting by k the area or the number of pixels of a screen element, only $k+1$ different intensity levels instead of 2^k levels are reproduced since the gray level in a screen element is determined only by the number of dots in the region. So, large screen element is required to have effective tone scale. On the contrary the size of a screen element should be small for effective resolution. This suggests a serious tradeoff between effective resolution and effective tone scale. The algorithm to be proposed in this paper resolves it by introducing adaptive mechanism to determine cluster sizes.

In most of the conventional cluster-dot halftoning algorithms the output image plane is partitioned into screen elements in a fixed manner independent of given input images. A key idea of our algorithm is to partition the output plane into screen elements of various sizes to reflect spatial frequency distribution of an input image. This adaptive method is a solution to balance effective resolution and effective tone scale in the following sense. The two indices are both important, but one is more important than the other depending on spatial frequency distribution of an input image. That is, resolution is more important in a high-frequency part to have a sharp contour, so that the sizes of screen elements should be kept small. On the other hand, tone scale is more meaningful in a low-frequency part with intensity levels changing smoothly, and so larger sizes of screen elements are preferred. All these requirements suggest the following geometric optimization problem. Given a continuous-tone image A and a scaling factor to define the size of an output image, we first compute spatial frequency distribution by applying Laplacian or Sobel differential operator. Then, each grid in the output image plane is associated with a disc of radius reflecting the Laplacian value at the corresponding point. Now, we have a number of discs of various radii. Then, the problem is to choose a set of discs to cover the output plane in an optimal way. The optimality criterion should reflect how large area is covered by exactly one disc from the set, which implies minimization of the area of unoccupied region and intersection among chosen discs to make the resulting screen elements rounded figures.

3 Problem Formulation

This section gives a formal definition of the problem. Input is an $M \times N$ matrix $R = (r_{ij}),\ 0 \le i < M, 0 \le j < N$ of positive real numbers. Each matrix entry r_{ij} specifies a radius of a disc to be placed at the corresponding position (i, j) in the plane. A matrix B of the same size is a binary matrix to be calculated. Each binary value b_{ij} is 1 when a disc of radius r_{ij} is placed at position (i, j) in the plane and 0 otherwise. Given two matrices R and B, the area covered by exactly one accepted disc, which is denoted by $g(R, B)$, is our objective function to be maximized. In other words, given a matrix R, we want to find a binary matrix B that maximizes the objective function $g(R, B)$.

We could also consider the problem by replacing discs by squaress. Then, each r_{ij} represents a side length of a square to be placed at (i, j). This problem is similar to that of packing n equal discs of largest possible radius in a unit square and to that of covering a unit square by n equal discs of smallest possible radius. In fact, if all the input values r_{ij} are equal, it becomes a discrete decision problem of the above problems in such a sense that disc center locations are restricted to grid points. The computational hardness of the problem is well recognized in the literature, but it is still open whether this problem is NP-hard or not.

4 An Efficient Algorithm for 1-D Version of the Problem

4.1 Graph-Based Approach

Let us first consider the one-dimensional version of the problem. We will show that an optimal solution is found in polynomial time in this case. We give an algorithm by reducing the problem to a longest path finding problem on a directed acyclic graph, which runs in $O(n^2)$ time and space. Later on, the space complexity will be improved. The idea of the algorithm is utilized in an approximation algorithm for the two-dimensional problem.

Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of n positive real numbers. For each r_i, we define an interval $I_i = [x_L(I_i), x_R(I_i)]$ where $x_L(I_i) = i - r_i$ and $x_R(I_i) = i + r_i$, referred to as the left and right endpoints of the interval I_i. Let $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ be a set of all such intervals. For a subset $\mathcal{I}'$ of $\mathcal{I}$ we define its gain $g(\mathcal{I}')$ by

$$g(\mathcal{I}') = \text{the total length of intervals covered exactly once by intervals in } \mathcal{I}'. \tag{1}$$

Given a set $\mathcal{I}$ of n intervals, our objective is to find a subset $\mathcal{I}^*$ of $\mathcal{I}$ maximizing the gain, the total length covered exactly once.

An example is shown in Fig. 1 with 12 intervals. Of course, there are 2^{12} different choices of intervals. When we choose three intervals $I_2 = [s_2, t_2], I_6 = [s_6, t_6]$, and $I_{10} = [s_{10}, t_{10}]$, the interval $[s_6 = 4.2, t_2 = 4.5]$ is doubly covered , and $[t_6 = 7.8, s_{10} = 8.0]$ and $[t_{10} = 12, t_{12} = 13]$ are empty. The remaining part is covered exactly once. Thus, the gain of this subset is given by $2r_2 + 2r_6 + 2r_{10} - 0.3 * 2 = 5 + 3.6 + 4 - 0.6 = 12$.

The following is a key observation to an efficient algorithm for the problem.

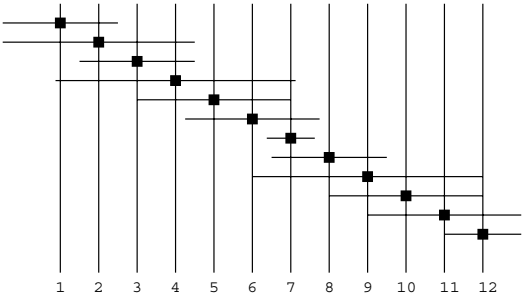


Fig. 1. An example of the problem: $r_1 = 1.5, r_2 = 2.5, r_3 = 1.5, r_4 = 3.1, r_5 = 2.0, r_6 = 1.8, r_7 = 0.7, r_8 = 1.6, r_9 = 3.0, r_{10} = 2.0, r_{11} = 2.0, r_{12} = 1.0$.

Lemma 1. *For any set $\mathcal{I}$ of intervals there is an optimal subset $\mathcal{I}^*$ of $\mathcal{I}$ such that no point is covered by three intervals from $\mathcal{I}^*$.*

A set $\mathcal{I}$ of intervals is called an *at most doubly overlapping interval set* if no three of them have non-empty intersection, or any point is covered by at most two intervals in $\mathcal{I}$. The lemma 1 guarantees that restriction to at most doubly overlapping interval sets does not lose all of optimal solutions. An at most doubly overlapping interval set can be expressed as a sequence of intervals $(I_{\sigma(1)}, I_{\sigma(2)}, \ldots, I_{\sigma(k)})$ such that

$$x_L(I_{\sigma(1)}) < x_L(I_{\sigma(2)}) < \cdots < x_L(I_{\sigma(k)})$$
$$I_{\sigma(i)} \cap I_{\sigma(j)} = \emptyset \text{ if } |i - j| \geq 2,$$

that is, one interval $I_{\sigma(i)}$ possibly overlaps the next interval $I_{\sigma(i+1)}$ and the previous $I_{\sigma(i-1)}$, but no others.

Lemma 2. *When an at most doubly overlapping interval set $\mathcal{I}$ is given as a sequence of intervals $(I_{\sigma(1)}, I_{\sigma(2)}, \ldots, I_{\sigma(k)})$, then the gain of $\mathcal{I}$ is given by*

$$g(\mathcal{I}) = \sum_{i=1}^{k} |I_{\sigma(i)}| - 2\sum_{i=1}^{k-1} |I_{\sigma(i)} \cap I_{\sigma(i+1)}|, \tag{2}$$

where $|I_{\sigma(i)}|$ is the length of interval $I_{\sigma(i)}$, i.e., $|I_{\sigma(i)}| = x_R(I_{\sigma(i)}) - x_L(I_{\sigma(i)})$, and $|I_{\sigma(i)} \cap I_{\sigma(i+1)}|$ is that of the intersection of the two consecutive intervals.

Proof. The total length of the union of all intervals is given by $\sum_{i=1}^{k} |I_{\sigma(i)}| - \sum_{i=1}^{k-1} |I_{\sigma(i)} \cap I_{\sigma(i+1)}|$. Since the intersection of consecutive intervals should be excluded from the singly-covered region, we have to reduce its length $\sum_{i=1}^{k-1} |I_{\sigma(i)} \cap I_{\sigma(i+1)}|$.

Now we can reduce our problem to that of finding a maximum-weight path in a directed acyclic graph defined as follows: Given a set $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ of

intervals, an interval traversing graph $G = (V, E, W)$ has vertices corresponding to those intervals and two special vertices s and t. Edge set is defined as follows.
(1) $(I_u, I_v) \in E,\ I_u, I_v \in \mathcal{I}$ if and only if (i) $v > x_R(I_u)$, (ii) $\frac{|I_u \cap I_v|}{\min\{|I_u|,|I_v|\}} < \frac{1}{2}$, and (iii) there is no I_w such that $u < w < v$ and $I_u \cap I_w = I_w \cap I_v = \emptyset$.
(2) $(s, I_u) \in E, I_u \in \mathcal{I}$ if and only if there is no I_v such that $v < u$ and $I_u \in I_v = \emptyset$, and
(3) $(I_u, t) \in E, I_u \in \mathcal{I}$ if and only if there is no I_v such that $v > u$ and $I_u \in I_v = \emptyset$.

Edge weights are defined as follows.
(4) $w(I_u, I_v) = |I_u| - 2|I_u \cap I_v|$ for each $(I_u, I_v) \in E$,
(5) $w(s, I_u) = 0$, and
(6) $w(I_u, t) = |I_u|$.

Lemma 3. *Let $r_{\max}$ be the maximum among $\{r_1, r_2, \ldots, r_n\}$. Then, outgoing degree of a vertex in an interval traversing graph G associated with a set of intervals defined by $\{r_1, r_2, \ldots, r_n\}$ is at most $3\lceil r_{\max} \rceil + 1$.*

The largest value $r_{\max}$ can be assumed to be $O(n)$ since otherwise the problem becomes trivial. If $r_{\max} = O(n)$ then the number of edges is $O(n^2)$. In many practical cases $r_{\max}$ is a constant independent of n and then we have only linear number of edges.

Let $\mathcal{I}'$ be a subset of the whole interval set $\mathcal{I}$. $\mathcal{I}'$ is *redundant* if there is an interval $I_u \in \mathcal{I} \backslash \mathcal{I}'$ such that I_u does not intersect any interval in $\mathcal{I}'$. Obviously, if $\mathcal{I}'$ is redundant then $\mathcal{I}'$ is not optimal since we can increase its gain by inserting an interval intersecting no interval in $\mathcal{I}'$. $\mathcal{I}'$ is also redundant if it contains two intervals I_u and I_v such that $|I_u \cap I_v| / \min\{|I_u|, |I_v|\} \geq 1/2$.

Lemma 4. *Let $\mathcal{I}$ be a given set of intervals and G be its associated interval traversing graph. Then, there is one-to-one correspondence between directed paths from s to t in G and at most doubly overlapping non-redundant interval sets. Furthermore, the sum of edge weights of such a path coincides with the gain of the corresponding set of intervals.*

Proof. Let $P = (s, I_{u_1}, I_{u_2}, \ldots, I_{u_k}, t)$ be any directed path from s to t in G. Then, $u_1 < u_2 < \cdots < u_k$ since $u_i > x_R(I_{u_{i-1}}) > u_{i-1}$ for $i = 2, 3, \ldots, k$. The set of intervals $\{I_{u_1}, I_{u_2}, \ldots, I_{u_k}\}$ is an at most doubly overlapping interval set. If three intervals have common intersection, one of their center points must be in the union of the other two intervals and thus it must be in one of the other intervals, which cannot happen by the definition of the graph. The interval set is not redundant again by the definition.

The proof for the other direction is similar. Lastly we can observe that the sum of edge weights of P and the gain of the corresponding interval set are both given by

$$\sum_{i=1}^{k} |I_{u_i}| - 2\sum_{i=1}^{k-1} |I_{u_i} \cap I_{u_{i-1}}|.$$

Theorem 1. *Given a set of n intervals associated with n real numbers $r_1, r_2, \ldots, r_n$, an optimal subset can be found in time $O(nr_{\max})$ as a maximum-weight path in the corresponding interval traversing graph, where $r_{\max}$ is the largest among $r_1, r_2, \ldots, r_n$.*

Proof. The lemma 4 guarantees that a maximum-weight path in the graph gives an optimal subset. Since the graph is a directed acyclic graph, such a path can be found in time linear in the number of edges, that is, in time $O(nr_{\max})$ by Lemma 3.

One disadvantage of the above-described approach is high space complexity. The number of edges is $O(n^2)$. Fortunately, we can reduce the space complexity while keeping the running time. An idea for the efficiency is dynamic programming combined with plane sweep paradigm.

4.2 Plane Sweep Approach

We sweep all the intervals, from left to right: Let $T = (t_{\sigma(1)}, t_{\sigma(2)}, \ldots, t_{\sigma(n)})$ be the increasing order of right endpoints (with larger coordinates) of n given intervals. Let $g(i)$ be the maximum gain of any interval subfamily from our intervals $\mathcal{I}$ that is contained in $(-\infty, t_{\sigma(i)}]$ where the gain is the total length covered exactly once.

Our algorithm to be presented is based on dynamic programming. Our goal is to compute the gain $g(n)$ at the rightmost endpoint $t_{\sigma(n)}$. We claim that starting with $g(0) = 0$, we can compute $g(1), g(2), \ldots, g(n)$ by

$$g(i) = \max \{\ g(i-1), \tag{3}$$

$$\max_{t_{\sigma(j)} < s_{\sigma(i)}} g(j) + t_{\sigma(i)} - s_{\sigma(i)}, \tag{4}$$

$$\max_{s_{\sigma(i)} \le t_{\sigma(j)} \le t_{\sigma(i)}} g(j) + t_{\sigma(i)} - s_{\sigma(i)} - 2(t_{\sigma(j)} - s_{\sigma(i)})\}. \tag{5}$$

Now, we shall prove that $g(i)$ correctly computes the gain of an optimal interval subfamily of $\mathcal{I}$, assuming for simplicity that all the interval endpoints are distinct.

For the interval $I_{\sigma(i)}$ there are two cases to consider. If it is not chosen as a member of an optimal solution when the search space is restricted to the left of $t_{\sigma(i)}$, the largest gain must have been achieved at some right endpoint of an interval. In this case the line (3) above guarantees the largest gain.

So, assume that the interval $I_{\sigma(i)}$ is a member of an optimal subfamily $\mathcal{I}^*$ restricted to the left of $t_{\sigma(i)}$. Let $I_{\sigma(j)}$ be the interval in $\mathcal{I}^*$ whose right endpoint $x_R(I_{\sigma(j)})$ is largest among those in $\mathcal{I}^*$ with $x_R(I_{\sigma(j)}) < x_R(I_{\sigma(i)})$. If the entire interval $I_{\sigma(j)}$ is located to the left of $I_{\sigma(i)}$, that is, if $x_R(I_{\sigma(j)}) \le x_L(I_{\sigma(i)})$, then the line (4) guarantees optimality of $g(i)$ under the inductive hypothesis that $g(j)$ achieves an optimal value if $j < i$.

Thus, the remaining case to consider is that the interval $I_{\sigma(j)}$ intersects $I_{\sigma(i)}$.
Case 1: The interval $I_{\sigma(j)}$ is properly contained in $I_{\sigma(i)}$.

In this case, $f(j,i) = g(j) + x_R(I_{\sigma(i)}) - x_L(I_{\sigma(i)}) - 2(x_R(I_{\sigma(j)}) - x_L(I_{\sigma(i)}))$ may not be a correct gain at $x_R(I_{\sigma(i)})$ since we may have some empty or doubly-covered interval in $[x_L(I_{\sigma(i)}), x_L(I_{\sigma(j)})]$. If we have such an empty interval in $[x_L(I_{\sigma(i)}), x_L(I_{\sigma(j)})]$, then $\mathcal{I}^*$ has another right endpoint $x_R(I_{\sigma(k)})$ in the interval or to the left of $x_L(I_{\sigma(j)})$ since otherwise $\mathcal{I}\backslash\{\mathcal{I}_{\sigma()}\}$ is better than $\mathcal{I}^*$, contradicting the optimality of $\mathcal{I}^*$. In the former case,

$f(k,i) = g(k) + x_R(I_{\sigma(i)}) - x_L(I_{\sigma(i)}) - 2(x_R(I_{\sigma(k)}) - x_L(I_{\sigma(i)}))$

gives a larger value than $f(j,i)$ and thus the gain $g(i)$ is correctly computed. On the other hand, in the latter case, that is, if we have a doubly-covered interval in $[x_L(I_{\sigma(i)}), x_L(I_{\sigma(j)})]$, that is, if we have two intervals $I_{\sigma(j)}$ and $I_{\sigma(k)}$ such that $x_L(I_{\sigma(i)}) < x_L(I_{\sigma(j)}) < x_R(I_{\sigma(k)}) < x_R(I_{\sigma(j)}) < x_R(I_{\sigma(i)})$, then the point $x_R(I_{\sigma(k)})$ is covered by the three intervals $I_{\sigma(i)}, I_{\sigma(j)}$, and $I_{\sigma(k)}$, which imply non-optimality of $\mathcal{I}^*$ due to Lemma 1, a contradiction. Thus, we do not need to worry about the case.

Case 2: The interval $I_{\sigma(j)}$ is not properly contained in $I_{\sigma(i)}$.

In this case the left endpoint $x_L(I_{\sigma(i)})$ is contained in the interval $I_{\sigma(j)}$. By Lemma 1 the interval $[x_L(I_{\sigma(i)}), x_R(I_{\sigma(j)})]$ is covered by $I_{\sigma(i)}$ and $I_{\sigma(j)}$ and no others in $\mathcal{I}^*$. Thus, $g(j)$ is correctly computed using the line (5).

5 2-Dimensional Problem: Problem Formulation and Approximation Algorithm

In this section we consider the original problem in the two dimensions. Let $R = (r_{ij})$ be a matrix of positive real numbers. For each (i,j) we define a disc C_{ij} of radius r_{ij} with its center at a lattice point (i,j). For a subset $\mathcal{S}'$ of the set $\mathcal{S}$ of all such discs we define its gain $g(\mathcal{S}')$ by

$$g(\mathcal{S}') = \text{the total area of regions covered exactly once by discs in } \mathcal{S}'. \quad (6)$$

Given a set $\mathcal{S}$ of discs, the problem is to find a subset $\mathcal{S}'$ of $\mathcal{S}$ maximizing the gain. This problem seems to be NP-hard although no proof is obtained.

Unfortunately, there is no such nice property stated in Lemma 1, that is, restriction to at most doubly overlapping disc sets may lose all of optimal solutions in this case. Our approximation algorithm has the performance ratio 5.83 and runs in $O(n \log n)$ time where $n = MN$ is the total number of discs.

To describe the algorithm we prepare some terminologies. C_u denotes a disc with its center at point u. $r(C_u)$ denotes the radius of a disc C_u. By $R_\rho(C_u)$ we denote a disc obtained by contracting C_u by a factor of $\rho, 0 < \rho < 1$, and therefore $r(R_\rho(C_{ij})) = \rho r(C_u)$. The contracted disc $R_\rho(C_u)$ is called the *core* of the disc. We say that a disc C_v *violates* another disc C_u if C_v intersects the core of C_u and C_v *safely intersects* C_u otherwise.

We start with the following simple approximation algorithm.

[Algorithm 1]
· Sort all the discs in the decreasing order of their radii.
· for each disc C_u in the order do
· if C_u does not intersect any previously accepted disc
· then accept C_u else reject C_u.
· Output all the accepted discs.

Lemma 5. *Algorithm 1 finds a 9-approximate solution.*

Proof. Let $\mathcal{S}'$ be a solution (set of discs) found by Algorithm 1. Then, all the discs in $\mathcal{S}'$ are disjoint, and thus it suffices to show that the area of the union of all given discs is at most 9 times larger than the total area covered by the discs in $\mathcal{S}'$. Let C_u be any given disc which is not accepted by the algorithm. Then, there must be some disc C_v which has been accepted before examining C_u and intersects C_u. Since discs are examined in the decreasing order of their radii, we have $r(C_u) \geq r(C_v)$. Thus, if we enlarge the accepted disc C_v by a factor of 3, then the rejected disc is completely contained in the enlarged disc. It implies that if we blow up ever accepted disc then it covers the union of all given discs.

The performance ratio can be improved to 5.83. The idea is to allow some overlap among accepted discs.

[Algorithm 2]
· Sort all the discs in the decreasing order of their radii.
· for each disc C_u in the order do{
· if C_u is not violated by any previously accepted disc,
· then accept C_u else reject C_u.
· Output all the accepted discs.

Lemma 6. *Algorithm 2 finds a* 5.83*-approximate solution.*

Proof. Let C_v be any disc rejected in the algorithm. Then, there must be a disc C_u such that
(1) C_u has been accepted before examining C_v, which implies $r(C_u) \geq r(C_v)$, and
(2) C_u violates C_v, that is, C_u intersects the core of C_v.

We enlarge each accepted disc C_u by a factor of $2 + \rho$ while keeping the center. The resulting disc coincides with the region swept by a copy of the disc C_u so that it touches the core boundary of C_u. Thus, the enlarged disc has radius $(2+\rho)r(C_u)$. It implies that if we enlarge every accepted disc by a factor of $(2 + \rho)$ then each rejected disc is contained in some such enlarged disc.

Next, we consider how much area is covered exactly once by accepted discs in each such enlarged disc region. For this purpose we take the multiplicatively weighted Voronoi diagram of the accepted discs, weighting each accepted disc by its radius. Then we truncate each Voronoi-cell of an accepted disc by its blown-up copy. We claim that for $\rho = \sqrt{2} - 1$ we get in each of these cells a ratio of singly covered area to total area of at least $\frac{1}{3+2\sqrt{2}} = 1/5.83...$, which implies the

same ratio for the whole set. To prove this, we notice that the cell is star-shaped, so it is sufficient to prove this for each small angular sector from the center of the disc C_u. Now if we look from the center in some direction, either the boundary of the cell in that direction is outside the disc C_u, then we know that it is not too far outside (at most a factor $2+\rho$ of the disc radius), and the disc cannot be overlapped by any other disc in this direction, so we have the ratio $1/(2+\rho)^2$ in an angular sector in that direction. Or the boundary of the cell is inside the disc C_u, in that case part of the disc C_u is overlapped by another disc, but at least we know that the core of C_u is not overlapped by another disc, so in that angular sector we have a ratio at least ρ^2.

The lower bound on the ρ values comes from the condition that the area of the disc C_u within the cell is at least that outside it. That is, we have $((1+\rho)/2)^2 \geq 1-((1+\rho)/2)^2$, and hence $\rho \geq \sqrt{2}-1$.

6 Heuristic Algorithms and Experimental Results

Now we propose heuristic algorithms. Although the algorithms have no theoretically guaranteed performance ratios, experimental results verify their effectiveness by their reasonable outputs. We have two heuristic algorithms. One is quite simple but very fast. The other takes some time but it produces better outputs because it is a natural extension of a polynomial-time exact algorithm for the one-dimensional version of the problem.

6.1 Heuristic Algorithm 1

The first heuristic algorithm is a simple greedy algorithm. We first fix a contraction factor ρ appropriately, say to 0.85 in the experiments. Then, we scan the matrix elements in a raster order. At each lattice point we accept the disc at the point if its core does not intersect the core of any previously accepted disc and reject it otherwise. If we can assume that the number of accepted disc centers in an influence region is bounded by some constant, then the algorithm runs in linear time. An influence region for a point (i,j) is a rectangular region with two corners at $(i-\rho(r_{ij}+r_{\max}), j-\rho(r_{ij}+r_{\max}))$ and $(i, j+\rho(r_{ij}+r_{\max}))$, where $r_{\max}$ is the maximum value of $r_{pq} \in R$. We can maintain a set of center points of accepted discs using some appropriate data structure such as a range tree. Because of the nature of the algorithm the first disc at the lower left corner is always accepted, which may be a bad choice for future selection.

We have implemented the Heuristic Algorithm 1 using LEDA 4.1. We applied it to sample images of sizes 106×85 and 256×320 each to be enlarged into those of 424×340 and 1024×1280, respectively. The running time is 0.06 seconds for the smaller image and 0.718 seconds for the larger one on a personal computer, DELL Precision 350 with Pentium 4. Fig. 2 illustrates an arrangement of contracted discs and its associated Voronoi diagram for the set of those center points.

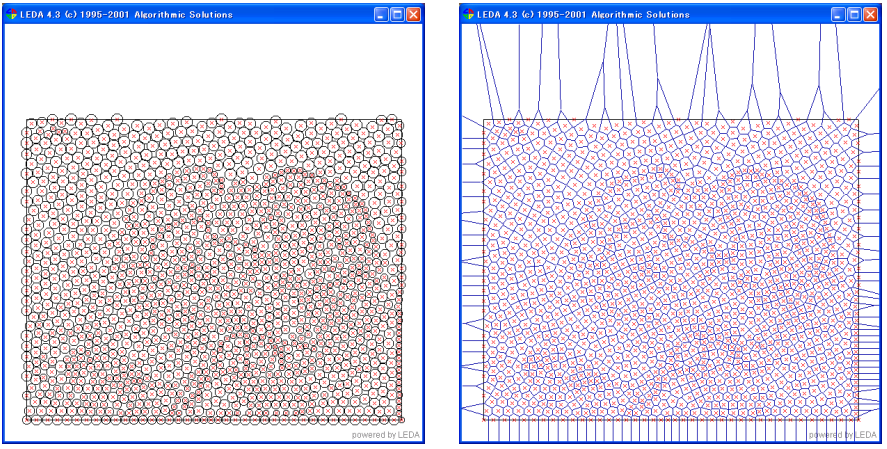


Fig. 2. An arrangement of discs accepted by Heuristic Algorithm 1 and resulting Voronoi diagram.

6.2 Heuristic Algorithm 2

The second heuristic is based on Algorithm 2 in the previous section, which examines discs in the decreasing order of their sizes and accepts discs if their core parts do not intersect the core part of any previously accepted disc. It took more time than Heuristic Algorithm 1 for all of our input images and the output quality was also worse. For example, it ran in 0.109 and 1.031 seconds for small and large input images, respectively and the gains achieved were 151474 and 1358940 for small and large images. The gains achieved by Heuristic 1 were 174022 and 1582940, respectively. Heuristic 1 achieved better results for all the test images than Heuristic 2.

6.3 Iterative Improvement

Set of discs obtained by heuristic algorithms can be improved by applying a standard iterative improvement strategy based on flipping of discs. That is, for each accepted disc we check whether more area can be covered by replacing it with neighboring discs. The running time of this iterative improvement depends on the number of flipping operations. Typical time for the improvement was 0.469 and 4.39 seconds for small and large images, respectively. Improvement over the area was roughly 1% in most cases.

6.4 Halftoning Based on Resulting Voronoi Diagram

Center points of accepted discs specify center points of screen elements for digital halftoning. Each Voronoi region in an associated Voronoi diagram is filled in according to the gray level at corresponding place.

7 Concluding Remarks

We have considered a geometric optimization problem to choose discs among n given discs to maximize the singly-covered area in conjunction with applications to adaptive digital halftoning. It finds a hard problem from a theoretical point of view, but a heuristic algorithm proposed in this paper works well for our practical applications.

References

1. T. Asano, T. Matsui, and T. Tokuyama: "Optimal Roundings of Sequences and Matrices," *Nordic Journal of Computing*, Vol.7, No.3, pp.241–256, Fall 2000.
2. K.J. Nurmela and P.R.J. Oestergard, "Packing up to 50 Equal Circles in a Square," Discrete Comput. Geom., 18:111–120, 1997.
3. V. Ostromoukhov: "Pseudo-Random Halftone Screening for Color and Black&White Printing," *Proceedings of the 9th Congress on Advances in Non-Impact Printing Technologies*, Yokohama, pp.579–581, 1993.
4. V. Ostromoukhov, R. D. Hersch: "Stochastic Clustered-Dot Dithering," *Journal of Electronic Imaging*, Vol.8, No.4, pp.439–445, 1999.
5. S. Sasahara T. Asano: "Adaptive Cluster Arrangement for Cluster-dot Halftoning Using Bubble Packing Method," *Proceeding of 7th Japan Korea Joint Workshop on Algorithms and Computation*, pp.87–93, Sendai, July 2003.

On Local Transformations in Plane Geometric Graphs Embedded on Small Grids*

(Extended Abstract)

Manuel Abellanas[1], Prosenjit Bose[2], Alfredo García[3], Ferran Hurtado[4], Pedro Ramos[5], Eduardo Rivera-Campo[6], and Javier Tejel[3]

[1] Facultad de Informática, U. Politécnica de Madrid, Madrid, Spain
[2] School of Computer Science, Carleton U., Ottawa, Canada.
[3] Fac. Ciencias. Dep. de Métodos Estadísticos, U. de Zaragoza, Zaragoza, Spain
[4] Dep. de Matemàtica Aplicada II, U. Politècnica de Catalunya, Barcelona, Spain
[5] Dep. de Matemáticas, U. de Alcalá, Madrid, Spain
[6] Dep. Matemáticas, U. Autónoma Metropolitana-Iztapalapa, México D.F., México

Abstract. Given two n-vertex plane graphs G_1 and G_2 embedded in the $n \times n$ grid with straight-line segments as edges, we show that with a sequence of $O(n)$ point moves (all point moves stay within a $5n \times 5n$ grid) and $O(n^2)$ edge moves, we can transform G_1 into G_2. In the case of n-vertex trees, we can perform the transformation with $O(n)$ point and edge moves, and show this is optimal. We also study the equivalent problems in the labelled setting.

1 Introduction

Informally, a local transformation is an operation performed on the vertices and edges of a graph. The term local is used because generally the operation does not affect the whole graph. Typically, the vertices of the graph affected by a local transformation are the neighborhood of a constant number of vertices. For example, an edge deletion and edge insertion are two local tranformations that affects the neighborhood of two vertices.

The local transformation that intiated this study is referred to as an *edge flip* or more generally an *edge move*. The class of graphs on which edge flips are defined is triangulations (and sometimes *near-triangulations*[1]) and edge moves are normally defined on planar graphs. The operation of an edge flip or edge

* This work was initiated when the authors were attending a workshop at the Universidad de Zaragoza. The second and sixth authors were on sabbatical leave at UPC. This work is partially supported by MCYT TIC02-4486-C02-1, SAB 2000-0234 grant of MECD Spain, a grant by Conacyt Mexico, a PIV 2001 grant of Generalitat de Catalunya, NSERC Canada, MCYT-FEDER BFM2002-0557, MCYT-FEDERBFM2003-0368, Gen. Cat 2001SGR00224, and DGA-2002-22861.

[1] A near triangulation is a plane graph where every face except possibly the outerface is a triangle.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 22–31, 2004.
© Springer-Verlag Berlin Heidelberg 2004

move is simply the deletion of an edge, followed by the insertion of another edge such that the resulting graph remains planar and simple.

Wagner[10] proved that given any two triangulations $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_1| = |V_2| = n$, there always exists a finite sequence of edge moves that transforms G_1 into a graph $G_3 = (V_3, E_3)$ that is isomorphic to G_2. That is, there exists a mapping $\phi : V_2 \to V_3$ such that for $u, v \in V_2$, $uv \in E_2$ if and only if $\phi(u)\phi(v) \in E_3$. Subsequently, Komuro[7] showed that in fact $O(n)$ edge flips suffice. Recently, Bose et al.[2] showed that $O(\log n)$ simultaneous edge flips suffice and are sometimes necessary. This setting of the problem is referred to as the combinatorial setting since the triangulations are only embedded combinatorially, i.e. the cyclic order of edges around each vertex is defined.

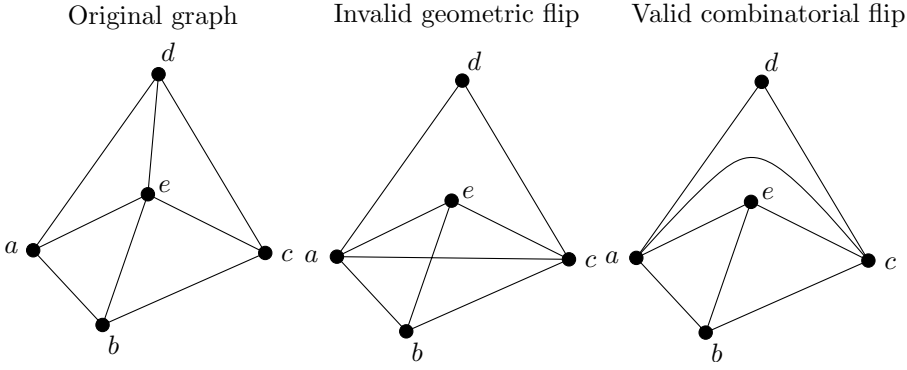


Fig. 1. Valid combinatorial edge flip of de but invalid geometric edge flip.

In the geometric setting, a triangulation or near-triangulation is embedded in the plane such that the vertices are points and the edges are straight-line segments. Henceforth, we only consider graphs embedded in the plane having straight-line segments for edges. Edge flips and edge moves are still valid operations in this setting, except that now the edge that is added must be a line segment and this line segment cannot properly intersect any of the existing edges of the graph. This additional restriction implies that there are valid edge moves in the combinatorial setting that are no longer valid in the geometric setting since eventhough the graph resulting after a move is planar and simple, it does not have a plane embedding. See Figure 1.

Lawson[8] showed that given any two near-triangulations N_1 and N_2 embedded on the same n points in the plane, there always exists a finite sequence of edge flips that transforms the edge set of N_1 to the edge set of N_2. Hurtado et al.[6] showed that $O(n^2)$ flips are always sufficient and sometimes necessary. Subsequently, Galtier et al.[5] showed that $O(n)$ simultaneous edge flips are sufficient and sometimes necessary.

Note that there is a discrepancy between the combinatorial result and the geometric one. In the combinatorial setting, Wagner[10] showed that *every* triangulation on n vertices can be attained from every other triangulation via edge flips. In the geometric setting, Lawson[8] showed that only the near-triangulations that

are defined on the specified point set can be attained via edge flips. For example, in the geometric setting, given a set of points in convex position, the only plane graphs that can be drawn without crossing are outer-planar.

It is precisely this discrepancy that sparked our investigation. The first question we asked is whether or not there exists a simple local transformation that permits the enumeration of all n-vertex triangulations in the geometric setting. In order to answer this question, the local transformation must be more general than an edge move. Two key ingredients need to be specified for this question. First, we need to specify the set of points P on which these graphs are embedded and on which the tranformations can be performed. To overcome the discrepancy with the combinatorial setting, this set of points must have the property that every n-vertex triangulation has a straight-line embedding on an n-point subset of P. Such a set of points is called a *universal point set*. Schnyder[9] showed that the $n \times n$ grid is a universal point set for all n-vertex planar graphs (see also [4]). Therefore, a grid is a natural choice for this setting. However, using a grid comes at a cost since there are many collinear points in a grid, we need to deal specifically with degeneracies. Despite this obstacle, we use grids as our universal point set and we outline the exact grid sizes required for our results. All of our grid sizes are within a constant of the optimal for straight-line embeddings of planar graphs. It is important to keep the grid size as small as possible since large grids hinder practical applications of these transformations.

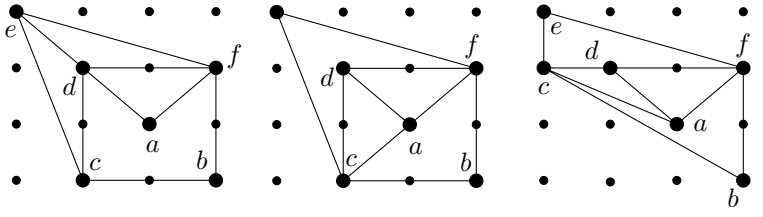


Fig. 2. Edge ed moved to ca followed by point move of c

The second ingredient is the set of allowable local transformations. Aside from the edge move, the other local transformation we use is a *point move*. A point move is simply the modification of the coordinates of one vertex. The move is valid provided that after moving the vertex to a new grid point, no edge crossings are introduced.

Due to space constraints, some proofs are omitted. Full proofs are available in the technical report and journal version of the paper.

2 Transforming One Plane Triangulation to Another

In this section, we show that $O(n)$ point moves and $O(n^2)$ edge moves suffice to transform one plane n-vertex triangulation into another. More precisely, let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_1| = |V_2| = n$ be two triangulations embedded in an $n \times n$ grid. Let the origin $(0, 0)$ of this grid be the bottom left

corner. Let P be a $5n \times 5n$ grid with bottom left corner located at $(-2n, -2n)$ and top right corner located at $(3n, 3n)$. During the whole sequence of moves, the location of every point move is a grid point of P (i.e. P is our universal point set). We show how to construct a sequence of $O(n^2)$ edge moves and $O(n)$ point moves that transforms both G_1 and G_2 into a canonical form. The canonical triangulation is a triangulation where the outer face consists of vertices located at $(-2n, -1), (3n, -1)$, and $(\lceil n/2 \rceil, 3n)$. The other $n-3$ vertices are located at $(\lceil n/2 \rceil, 3n - i), 1 \leq i \leq n-3$. The two bottom corner vertices are stars adjacent to all other vertices and the graph induced by the remaining vertices is a path which we will call the *spine*.

Before showing how to construct the sequence of point and edge moves, we need to establish a few basic building blocks.

Lemma 1. *[6] Let $T = (V, E)$ be an arbitrary near-triangulation whose vertex set is a set of n points in the plane. Let a, b, c be three consecutive vertices on the outerface of T. Let P be the path from a to c on the convex hull of $V \setminus b$. With precisely k edge moves, where k is the number of edges of T that intersect P, we can transform T into a triangulation that contains P. Note that k is $O(n)$.*

Observation 1 *Let $\triangle(abc)$ be a triangle and x a point contained in its interior. Any line through x with both b and c in one half-plane must have a in the other and must intersect the line segments ab and ac.*

Observation 2 *Let $a = (0, 0)$, $b = (x_1, y_1)$ with $x_1, y_1 > 0$, $c = (x_1, y_1 + 1)$ and $d = (x_2, y_2)$ with $x_2 > x_1$ and $y_2 > (y_1 + 1)x_2/x_1$. The point c is contained in the interior of triangle $\triangle(abd)$.*

We now describe a sequence of edge moves and one point move which we will call an *apex slide*. The setting for an apex slide is the following. Let a, b, c be the vertices of the outerface of a triangulation G. Let x be a point such that xbc forms a triangle, both x and a are on the same side of a half-plane defined by the line through bc, and all other vertices of G are in $\triangle(abc) \cap \triangle(xbc)$.

Lemma 2. *With $O(n)$ edge moves and one point move, vertex a can be moved to point x.*

Proof. Let $C = c_0, c_1, \ldots, c_k$ be the clockwise order of the vertices of convex hull of $G \setminus a$ starting at $b = c_0$ and ending at $c_k = c$. By Lemma 1, with $O(n)$ edge moves, we can convert the triangulation contained in abc to one which contains the segments ac_i and the edges of C. Once this is accomplished, Observation 1 implies that we can move a to x without introducing any crossing since C is contained in both $\triangle(abc)$ and $\triangle(xbc)$. Thus, a total of $O(n)$ edge moves and 1 point move suffice as required.

To initiate the whole process, we need to show how given a triangulation embedded in the $n \times n$ grid we can always move the vertices of its outerface to the coordinates $(-2n, -1)$, $(3n, -1)$ and $(\lceil n/2 \rceil, 3n)$. We note that $O(n)$ edge moves and at most 8 point moves suffice to pull out the three vertices of the outerface into these three positions.

Lemma 3. *Given a triangulation $G = (V, E)$ embedded in the $n \times n$ grid, with $O(n)$ edge moves and at most 8 point moves, we can transform it into a triangulation whose outerface has coordinates $(-2n, -1)$, $(3n, -1)$ and $(\lceil n/2 \rceil, 3n)$. All other vertices of G have coordinate values between 0 and n (i.e. they are in the original $n \times n$ grid).*

We now describe the main step in the process. Let a, b, c be the vertices of the outerface of a triangulation G embedded on a grid, such that b and c lie on the same horizontal grid line L_1, there are at least $5n - 1$ grid points between b and c, the vertex a is above b and c, and a lies on a vertical line L_2 such that there are at least $2n$ grid points on segment between b and $z = L_1 \cap L_2$ and at least $2n$ grid points on the segment between c and z. Let x be a point of the grid that is not a vertex of G such that a and x lie on the same vertical grid line L_2. The triangle $\triangle(abc)$ is the outerface of G and the point x be strictly inside triangle $\triangle(abc)$. All other vertices of G are strictly inside triangle $\triangle(bxc)$. There are at least n grid points on the segment ax.

Lemma 4. *With $O(n^2)$ edge moves and $O(n)$ point moves, we can transform G into canonical form.*

Proof. We proceed by induction on the number h of vertices of G in $\triangle(bxc)$.

Base Case: $h = 0$. The lemma holds trivially since no moves are required.

Inductive Hypothesis: $0 \leq h \leq k, k > 0$. Assume that $d_1 h^2$ edge moves and $d_2 h$ point moves suffice with constants d_1 and d_2.

Inductive Step: $h = k + 1$. Let r be the first unoccupied grid point below a. Let $C = c_0, c_1, \ldots, c_{m+1}$ be the clockwise order of the convex hull of the vertices of $G \setminus a$ starting at $b = c_0$ and ending at $c = c_{m+1}$. Apply Lemma 1, to convert G to a triangulation containing C and all segments ac_i for vertices c_i of the convex hull. This is accomplished with $d_3 k$ edge moves for constant d_3.

Let $c_j c_{j+1}$ be the edge of the convex hull that intersects the vertical line through ax. If the line through ax contains a vertex of the convex hull, assume this vertex is c_j. There are two cases to consider.

Case 1. One of c_j or c_{j+1} is a vertex of the convex hull Assume without loss of generality that c_j is a convex hull vertex. Since c_j is a vertex of the convex hull, the points c_{j-1}, c_j, and c_{j+1} are not collinear. Since the grid point r is in triangle $\triangle(ac_{j-1}c_{j+1})$ by construction, we can apply an apex move to move point c_j to r.

Case 2. The edge $c_j c_{j+1}$ is in the interior of the convex hull edge $e = c_s c_t$ Assume for the moment that the edge e has positive slope. Since c_s is a vertex of the convex hull, this means that c_{s-1}, c_s and c_{s+1} are not collinear. By Observation 2, there is a grid point y one unit vertically above c_s inside triangle $\triangle(ac_{s-1}c_s)$. Apply an apex move to move c_s to y. This removes the collinearity from the convex hull. Now the edge yc_t is on the convex hull. Recompute the convex hull and apply Lemma 1 so that a is adjacent to all edges of the convex hull. Now

we have reduced the situation back to the previous case. A symmetric argument holds if e has negative slope.

Therefore, with d_4k edge moves and at most 2 point moves, we remove one vertex of G from $\triangle(bxc)$, and move it to r. Now, there are only k vertices of G remaining in the triangle $\triangle(bxc)$. Apply Lemma 1 so that r is adjacent to all vertices on the convex hull of $G \setminus \{a, r\}$. We can now apply the inductive hypothesis. The total number of edge moves is $d_1k^2 + d_4k$ and the total number of point moves is d_2k+2. If we set $d_1 > d_4$ and $d_2 > 2$, then $d_1k^2+d_4k < d_1(k+1)^2$ and $d_2k + 2 < d_2(k + 1)$.

Theorem 1. *Given an n-vertex triangulation $G = (V, E)$ embedded in the $n \times n$ grid with straight-line segments as edges, with $O(n^2)$ edge moves and $O(n)$ point moves (all point moves stay within the grid [(−2n, −2n),(3n, 3n)]), we can transform G into the canonical triangulation.*

Proof. Let R represent the points of the $n \times n$ grid containing G and let P represent the universal point set. First apply Lemma 3 to G. Then, we can apply Lemma 4. The theorem follows.

Corollary 1. *Given two n-vertex triangulations $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ embedded in the $n \times n$ grid with straight-line segments as edges, with $O(n^2)$ edge moves and $O(n)$ point moves (the [(−2n, −2n),(3n, 3n)] grid contains all point moves), we can transform G_1 into G_2*

Remark 1. We note that with a little care, our grid size can be reduced to $3n \times 3n$ at the expense of simplicity of exposition. We chose to keep the explanations simple in order to easily convey the main ideas rather than get bogged down in details.

3 Transforming One Tree to Another

In this section, we show that $O(n)$ point and edge moves suffice to transform one tree into another and this is optimal as there are pairs of trees that require $\Omega(n)$ point and edge moves to transform one into the other.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two trees embedded in the plane on an $n \times n$ grid with $|V_1| = |V_2| = n$. Let the origin $(0, 0)$ of this grid be the bottom left corner. Let P be an $n + 1 \times n + 1$ grid with bottom left corner located at $(-1, -1)$. During the whole sequence of moves, the location of every point move is a grid point of P. The approach is similar to that used for triangulations, but since trees are a simpler structure, the number of moves and the grid size are reduced.

Avis and Fukuda[1] showed that given any tree embedded in the plane, with at most $n - 1$ edge moves, this tree can be transformed into a canonical tree. However, their result does not hold in the presence of collinearities. We modify

their result to account for degeneracies. The canonical tree we strive for is the following. Let $p_1, p_2, \ldots, p_n$ be the vertex set of the given tree T. Relabel the points in the following manner. Let p_1 be the leftmost, bottommost point. Label the other points $p_2, \ldots, p_n$ in sorted order counter-clockwise around p_1 so that p_1p_2 and p_1p_n are on the convex hull, and if p_1, p_i, p_j are collinear, then $i < j$. The canonical tree is the following: the edge p_1p_i is in the tree if there is no point $p_j, j \neq i$ in the interior of the segment p_1p_i. If the segment p_1p_i has points in its interior, let p_k be the interior point closest to p_i. The segment p_kp_i is in the tree. Note that essentially this builds paths of collinear vertices from p_1.

Lemma 5. *A tree T with n-vertices embedded in the $n \times n$ grid can be transformed into the canonical tree with $n-2$ edge moves. Each edge move is planar.*

Proof. Let T be the given tree embedded on the points $p_1, \ldots, p_n$ labelled as above. Call an edge p_ip_j of T a *transversal edge* if the line through p_ip_j does not contain p_1. We proceed by induction on the number t of transversal edges.

Base Case: $t = 0$. In this case, T is the canonical tree.

Inductive Hypothesis: $t < k, k > 0$. With t edge moves, T can be transformed into the canonical tree.

Inductive Step: $t = k$. There always exists a transversal edge p_ip_j such that for any point p in the interior of segment p_ip_j, the segment p_1p does not intersect any other transversal edge. Removing p_ip_j disconnects T into two components C_1 containing p_i and C_2 containing p_j. Without loss of generality, let p_1 be in C_1. Let $p_1 = x_1, x_2, \ldots, x_a = p_j$ be the vertices of T on the segment p_1p_j. Since $p_1 \in C_1$ and $p_j \in C_2$, there exists a k such that $x_k \in C_1$ and $x_{k+1} \in C_2$. Add edge x_kx_{k+1} to the tree. Since we have reduced the number of transversal edges with one edge move, the result follows by induction.

Lemma 5 gives us the freedom to move to any tree defined on a given point set with $n-2$ edge moves. Given an n-vertex tree T embedded in the $n \times n$ grid, we show how to transform it into a path embedded on vertices $(-1, -1+i)$, $0 \leq i \leq n-2$. Let $p_1, p_2, \ldots, p_n$ be the n points of T. Relabel these points so that they are sorted by increasing X coordinate with p_1 being the leftmost, bottomost point. If two points p_i and p_j are on the same vertical grid line, then $i < j$ if p_i is below p_j. Now Lemma 5 implies that T can be transformed to the path $p_1, p_2, \ldots, p_n$ with $2n-2$ edge moves. We call such a path a *monotone* path

Lemma 6. *A monotone path embedded on the $n \times n$ grid can be transformed to the canonical path embedded on vertices $(-1, -1+i)$, $0 \leq i \leq n-2$ with n point moves.*

Proof. By definition, the half-plane to the left of the vertical line through the leftmost point is empty. Therefore, the leftmost, bottommost point can be moved to any grid point to its left. Move it to $(-1, -1)$. Once this point is moved, the next leftmost, bottommost point can be moved to $(-1, 0)$. The lemma follows by induction.

Theorem 2. *Given two trees T_1 and T_2 embedded on the $n \times n$ grid, with at most $4n-4$ edge moves and $2n$ point moves (where the point moves are restricted to remain in an $n+1 \times n+1$ grid), T_1 can be converted to T_2.*

Proof. The theorem follows from the discussion above and Lemmata 6 and 5.

In order to show the lower bound, take an n-vertex star and an n-vertex path each embedded on n different grid points. To convert the path to a star, we need at least $n-3$ edge moves since all vertices of the path have degree at most 2 and the star has a vertex of degree $n-1$. Similarly, since none of the points of the star coincide with the points of the path, we need at least n point moves to get from the vertex set of the path to that of the star.

Theorem 3. *There exist pairs of trees T_1 and T_2 embedded on the $n \times n$ grid that require at least $n-3$ edge moves and at least n point moves to transform one to the other.*

Remark 2. With a little care, we can use the same $n \times n$ grid as the one on which the tree is embedded, again at the expense of simplicity of exposition.

4 Transforming One Plane Graph to Another

We now show how to generalize the results from Section 2 to plane graphs. Given two plane graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ embedded in the $n \times n$ grid with $|V_1| = |V_2| = n$ and $|E_1| = |E_2| = m$, we show how to transform G_1 into G_2. We will assume that both graphs are connected. The obvious approach is to add *dummy edges* to both graphs until they are triangulations. Then, apply the previous result, ignoring the moves concerning dummy edges. Some details need to be outlined with this approach. For example, it is no longer clear what the canonical form is when the input graph is not a triangulation.

We first show how to transform G_1 into a canonical form. The problem is that since G_1 is not a triangulation, we need to specify precisely what the canonical form is. Recall the canonical form for triangulations and label its vertices in the following way. Let p_1 and p_2 be the left and right corners of the outerface and let p_3 be the apex. Label the vertices $p_4, \ldots, p_n$ in descending order on the spine from p_3. Label the edges adjacent to p_1 by $e_0, \ldots, e_{n-2}$ in clockwise order around p_1 with $e_0 = p_1p_3$ and $e_{n-2} = p_1p_2$. Label all the edges adjacent to p_2 except edge p_1p_2 and p_2p_3 by $e_{n-1}, \ldots, e_{2n-5}$, in counter-clockwise order with $e_{n-1} = p_2p_4$ and $e_{2n-5} = p_2p_n$. The value of m determines the shape of the canonical graph. Since G_1 is planar, $n-1 \leq m \leq 3n-6$. If $m = n-1$, then the canonical graph is a tree formed by the path from p_3 to p_n along with the edges p_1p_3 and p_2p_3. If $m > n-1$, let $k = m-n+1$. Augment the canonical tree with the edges $e_1, \ldots, e_k$.

The first step is to triangulate G_1. Bicolor the edges red and blue so that the original m edges are red and all additional edges are blue. By applying Theorem 1, we achieve a bicolored triangulation in canonical form. The red edges in this triangulation may not be in canonical form. An extra linear number of edge moves gives the resulting canonical graph for red edges.

Theorem 4. *Given two n-vertex plane graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ embedded in the $n \times n$ grid each having m edges, with $O(n^2)$ edge moves and $O(n)$ point moves (all point moves stay within the grid $[(-2n, -2n), (3n, 3n)]$), we can transform G_1 into G_2*

One aspect of this approach which is unsatisfactory is that throughout the sequence, the graph may become disconnected eventhough we start with a connected graph. This begs the question: is there a way to guarantee that in converting one plane graph into another, we remain connected throughout the whole sequence? We answer this in the affirmative. Note that point moves do not change the connectivity of a graph. Therefore, we solely need to concentrate on edge moves. The key idea is to maintain a connected spanning red graph after every edge move.

Lemma 7. *Let G be an n-vertex near-triangulation. Let m of the edges of G be colored red such that the graph induced by the red edges is connected and spanning. Let the remaining edges of G be colored blue. Let e be an edge of G to be flipped. With at most 1 edge move, we can flip edge e such that after each of the edge move and edge flip, the graph induced by the red edges remains connected and spanning.*

Proof. Let R be the graph induced by the m red edges. We need to show that we can flip an edge e of G such that R remains connected after the flip. Let e be the edge to be flipped. If e is blue, then flipping e does not affect the connectivity of the graph induced on the red edges. If e is red, then the only way that the connectivity of R is affected is if e is a cut edge[2] of R.

Since e is in G, e is adjacent to at least one triangular face of G. Let a, b, c with $e = ab$ be the three vertices defining this face. The edges bc and ac cannot both be red since this would contradict the fact that e is a cut edge. Since e is a cut edge, the deletion of e from R disconnects the graph into two components with a and b going to different components. Without loss of generality, assume that b and c are in different components. Then performing an edge move in the red graph from $e = ab$ to bc, we have a new set of m red edges that form a connected graph. Essentially, this amounts to coloring e blue and bc red. Now, since e is blue, we can flip e without affecting the connectivity of R.

Therefore, after one edge move, we can perform the flip. The lemma follows.

Since Lemma 1 uses edge flips in a near-triangulation, and these are the only edge moves we use, we conclude with the following.

Corollary 2. *Given two n-vertex plane graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ embedded in the $n \times n$ grid each having m edges, with $O(n^2)$ edge moves and $O(n)$ point moves (all point moves stay within the grid $[(-2n, -2n), (3n, 3n)]$), we can transform G_1 into G_2 while remaining connected throughout the sequence of moves.*

[2] A cut edge is an edge whose deletion disconnects a graph.

5 Labelled Transformations

In the labelled setting, we are given two plane graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, and a mapping $\phi : V_1 \to V_2$. Now perform a sequence of edge and point moves that transforms G_1 into a graph $G_3 = (V_3, E_3)$ that is isomorphic to G_2. This defines a mapping $\delta : V_1 \to V_3$. In the unlabelled case, we simply want G_3 to be isomorphic to G_2. In the labelled case, in addition, we want for every vertex $x \in V_1$, that $\phi(x) = \delta(x)$.

The same problems can be solved for trees, triangulations and plane graphs in the labelled setting. We simply state the theorems without proof due to space constraints.

Theorem 5. *Given two trees T_1 and T_2 embedded on the $n \times n$ grid, and a mapping ϕ of the vertices of T_1 to the vertices of T_2, with $O(n)$ point and edge moves, T_1 can be converted to T_2 respecting the mapping.*

Theorem 6. *Given two n-vertex plane graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ embedded in the $n \times n$ grid each having m edges, and a mapping ϕ of the vertices of G_1 to the vertices of G_2, with $O(n^2)$ edge moves and $O(n^2)$ point moves (all point moves stay within the grid $[(-2n, -2n), (3n, 3n)]$), we can transform G_1 into G_2 while respecting the given mapping and remaining connected throughout the sequence of moves.*

References

1. D. Avis and K. Fukuda, Reverse search for enumeration. *Discrete Applied Math.*, **65**:21–46, 1996.
2. P. Bose, J. Czyzowicz, Z. Gao, P. Morin, and D. R. Wood, Parallel diagonal flips in plane triangulations. Tech. Rep. TR-2003-05, School of Computer Science, Carleton University, Ottawa, Canada, 2003.
3. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Germany, 2nd edn., 2000.
4. H. de Fraysseix, J. Pach, and R. Pollack, How to draw a planar graph on a grid. *Combinatorica*, **10(1)**:41–51, 1990.
5. J. Galtier, F. Hurtado, M. Noy, S. Pérennes, and J. Urrutia, Simultaneous edge flipping in triangulations. *Internat. J. Comput. Geom. Appl.*, **13(2)**:113–133, 2003.
6. F. Hurtado, M. Noy, and J. Urrutia, Flipping edges in triangulations. *Discrete Comput. Geom.*, **22(3)**:333–346, 1999.
7. H. Komuro, The diagonal flips of triangulations on the sphere. *Yokohama Math. J.*, **44(2)**:115–122, 1997.
8. C. Lawson, Software for c_1 surface interpolation. In J. Rice, ed., *Mathematical Software III*, pp. 161–194, Academic Press, New York, 1977.
9. W. Schnyder, Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Symp. on Discrete Algorithms*, pp. 138–148, 1990.
10. K. Wagner, Bemerkung zum Vierfarbenproblem. *Jber. Deutsch. Math.-Verein.*, **46**:26–32, 1936.

Reducing the Time Complexity of Minkowski-Sum Based Similarity Calculations by Using Geometric Inequalities

Henk Bekker and Axel Brink

Institute for Mathematics and Computing Science, University of Groningen,
P.O.B. 800 9700 AV Groningen, The Netherlands.
bekker@cs.rug.nl, a.a.brink@student.rug.nl

Abstract. The similarity of two convex polyhedra A and B may be calculated by evaluating the volume or mixed volume of their Minkowski sum over a specific set of relative orientations. The relative orientations are characterized by the fact that faces and edges of A and B are parallel as much as possible. For one of these relative orientations the similarity measure is optimal. In this article we propose and test a method to reduce the number of relative orientations to be considered by using geometric inequalities in the slope diagrams of A and B. In this way the time complexity of $O(n^6)$ is reduced to $O(n^{4.5})$. This is derived, and verified experimentally.

1 Introduction: Minkowski-Sum Based Similarity Measures

Because shape comparison is of fundamental importance in many fields of computer vision, in the past many families of methods to calculate the similarity of two shapes have been proposed. Well-known families are based on the Hausdorff metric, on contour descriptors and on moments of the object, see [1] for an overview. Recently, a new family of methods has been introduced, based on the Brunn-Minkowki inequality and its descendants. The central operation of this method is the minimization of a volume or mixed volume functional over a set of relative orientations [2]. It is defined for convex objects, and can be used to calculate many types of similarity measures. Moreover, it is invariant under translation and rotation, and when desired, under scaling and reflection. The methods may be used in any-dimensional space, but we will concentrate on the 3D case. Experiments with these methods have been performed on 2D polygons and 3D polyhedra [3,4], and show that for polygons the the time consumption is low. However, already for 3D polyhedra of moderate complexity in terms of the number of faces, edges and vertices the time consumption is prohibitive. In this article we present a method to reduce the time complexity of these calculations by reducing the number of relative orientations to be considered.

The structure of this article is as follows. In this section we introduce the Minkowski sum, the notion of mixed volume, the Brunn-Minkowski inequalities,

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 32–41, 2004.
© Springer-Verlag Berlin Heidelberg 2004

and derive some example similarity measures. In section two we introduce the slope diagram representation of convex polyhedra, define the set of critical orientations to be considered, present the current algorithm to calculate a similarity measure, and discuss its time complexity. In section three we introduce and test the new and more efficient algorithm, and we derive its theoretical time complexity.

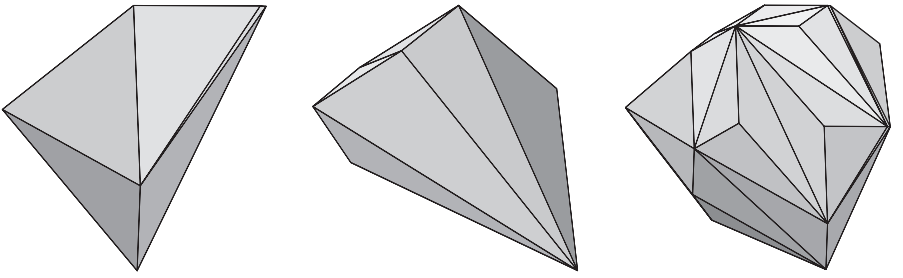


Fig. 1. Two polyhedra A and B and their Minkowski sum C. C is drawn on half the scale of A and B.

Let us consider two convex polyhedra A and B in 3D. The Minkowski sum C of two polyhedra A and B is another polyhedron, generally with more faces, edges and vertices than A and B, see figure 1. It is defined as

$$C \equiv A \oplus B \equiv \{a + b \mid a \in A, b \in B\}. \tag{1}$$

This definition does not give much geometrical insight how C is formed from A and B. To get some feeling for that, we separately look at two properties of C, namely its shape and its position. The shape of C may be defined by a sweep process as follows. Choose some point p in A, and sweep space with translates of A such that p is in B. C consists of all points that are swept by translates of A. The same shape C results when A and B are interchanged. The position of C is roughly speaking the vectorial sum of the positions of A and B. More precise, the rightmost coordinate of C is the sum of the rightmost coordinates of A and B, and analogously the leftmost, uppermost and lowermost coordinates of C. In this article only the shape of C plays a role, not its position. Obviously, the shape and volume of C depend on the relative orientation of A and B.

The volume of C may be written as

$$V(C) = V(A \oplus B) = V(A) + 3V(A, A, B) + 3V(A, B, B) + V(B). \tag{2}$$

Here, $V(A)$ and $V(B)$ are the volumes of A and B, and $V(A, A, B)$ and $V(A, B, B)$ are the *mixed volumes*, introduced by Minkowski [6]. Geometrically it is not obvious how the volume of A and B and the mixed volumes add up to the volume of C. However, it can be shown that $V(A, A, B)$ is proportional to the area of A and the linear dimension of B, and $V(A, B, B)$ is proportional to the linear dimension of A and the area of B.

As an example we derive two typical similarity measure expressions, based on the following two theorems [3,6]:

Theorem 1: For two arbitrary convex polyhedra A and B in R^3,

$$V(A, A, B)^3 \geq V(A)^2 V(B) \tag{3}$$

with equality if and only if $A = B$.

Theorem 2: For two arbitrary convex polyhedra A and B in R^3,

$$V(A \oplus B) \geq 8V(A)^{\frac{1}{2}} V(B)^{\frac{1}{2}} \tag{4}$$

with equality if and only if $A = B$.

From these theorems the similarity measures σ_1 and σ_2 respectively may be derived in a straightforward way,

$$\sigma_1(A, B) \equiv \max_{R \in \mathcal{R}} \frac{V(A)^{2/3} V(B)^{1/3}}{V(R(A), R(A), B)} \tag{5}$$

$$\sigma_2(A, B) \equiv \max_{R \in \mathcal{R}} \frac{8V(A)^{\frac{1}{2}} V(B)^{\frac{1}{2}}}{V(R(A) \oplus B)}. \tag{6}$$

Here $\mathcal{R}$ denotes the set of all spatial rotations, and $R(A)$ denotes a rotation of A by R. Because the volumes in these equations are always positive, σ_1 and σ_2 are always positive and ≤ 1, with equality if and only if $A = B$. Besides the inequalities in theorem1 and theorem2 many other inequalities exist, some based on the volume of the Minkowski sum, some on the mixed volume, some on the area of the Minkowski sum or the mixed area. From every of these inequalities a similarity measure may be derived. In this article we concentrate on computing σ_1 because the technique presented in this article to speed up this computation may be applied to other Minkowski sum based similarity calculations as well.

2 Calculating the Similarity Measure Straightforward

To find the maximum in (5), in principle an infinite number of orientations of A have to be checked. That would make this similarity measure useless for practical purposes. Fortunately, as is shown in [3], to find the maximum value only a finite number of relative orientations of A and B have to be checked. Roughly speaking these orientations are characterized by the fact that edges of B are as much as possible parallel to faces of A. To formulate this more precise we use the slope diagram representation (SDR) of polyhedra.

We denote face i of polyhedron A by $F_i(A)$, edge j by $E_j(A)$, and vertex k by $V_k(A)$. The SDR of a polyhedron A, denoted by SDR(A), is a subdivision on the unit sphere. A vertex of A is represented in SDR(A) by the interior of a spherical polygon, an edge by a spherical arc of a great circle, and a face by a vertex of a spherical polygon, see figure 2. To be more precise:

- *Face representation.* $F_i(A)$ is represented on the sphere by a point $SDR(F_i(A))$, located at the intersection of the outward unit normal vector u_i on $F_i(A)$ with the unit sphere.

- *Edge representation.* An edge $E_j(A)$ is represented by the arc of the great circle connecting the two points corresponding to the two adjacent faces of $E_j(A)$.
- *Vertex representation.* A vertex $V_k(A)$ is represented by the interior of the polygon bounded by the arcs corresponding to the edges of A meeting at $V_k(A)$.

Some remarks. From this description it can be seen that the graph representing SDR(A) is the dual of the graph representing A. SDR(A) is not a complete description of A, it only contains angle information about A. Obviously, when A is rotated by a rotation R, the slope diagram representation rotates in the same way, i.e., $SDR(R(A)) = R(SDR(A))$. In the following, when speaking about distance in an SDR we mean spherical distance, i.e. the length of an arc on the unit sphere. Because the angle between two adjacent faces of a polyhedron is always $< \pi$, the length of the arcs in a SDR is always $< \pi$.

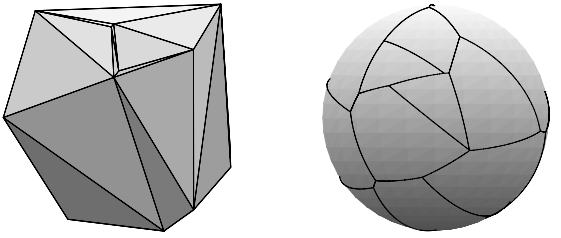


Fig. 2. (a): A polyhedron A. (b): The slope diagram representation of A. The orientations of A and $SDR(A)$ are the same, so with some patience it should be possible to see how they are related.

The slope diagram representation is useful to represent situations where faces and edges of A are parallel to faces and edges of B. It is easily verified that the faces $F_i(A)$ and $F_j(B)$ are parallel when in the overlay of $SDR(A)$ and $SDR(B)$ the point $SDR(F_i(A))$ coincides with the point $SDR(F_j(B))$. Also, an edge $E_i(B)$ is parallel to $F_j(A)$ when the point $SDR(F_j(A))$ lies on the arc $SDR(E_i(B))$. The description given earlier, stating that (5) obtains its maximum value when edges of B are as much as possible parallel to faces of A can now be made more precise in terms of their slope diagrams:

Theorem 3: When σ_1 is maximal then three points of $SDR(R(A))$ lie on three arcs of $SDR(B)$.

This theorem is derived in [3]. Unfortunately, this theorem does not tell for which three points in $SDR(R(A))$ and which three arcs in $SDR(B)$ σ_1 is maximal, thus to find the maximum, all rotations R have to be considered for which three points of SDR(R(A)) lie on three arcs of $SDR(B)$. So, for three given points p_1, p_2, p_3 in $SDR(A)$ and three arcs a_1, a_2, a_3 in $SDR(B)$, an algorithm is needed that calculates a spatial rotation R for which holds that $R(p_1)$ lies on

a_1, $R(p_2)$ lies on a_2 and $R(p_3)$ lies on a_3. We developed such an algorithm [5], and implemented it in the function tvt(). It takes as argument three points and three arcs and calculates a rotation R. It is called as tvt(p1, p2, p3, a1, a2, a3, R). The function tvt() first calculates a rotation R with the property that $R(p_1)$ lies on c_1, $R(p_2)$ lies on c_2 and $R(p_3)$ lies on c_3, where c_1, c_2, c_3 is the great circle carrying the arc a_1, a_2, a_3 respectively. When $R(p_1)$ lies on a_1, $R(p_2)$ lies on a_2 and $R(p_3)$ lies on a_3, tvt() returns "true", else it returns "false". The time complexity of tvt() is constant. Notice that the rotation returned by the call tvt(p1, p2, p3, a1, a2, a3, R), is the same as the rotation returned by the calls tvt(p1, p3, p2, a1, a3, a2, R), tvt(p2, p1, p3, a2, a1, a3, R), tvt(p3, p1, p2, a3, a1, a2, R), tvt(p3, p2, p1, a3, a2, a1, R) and tvt(p2, p3, p1, a2, a3, a1, R). That is because the the order of the statements "$R(p_1)$ lies on a_1, $R(p_2)$ lies on a_2, $R(p_3)$ lies on a_3" is irrelevant. In the implementation this observation may be used to gain a factor of six. Now calculating $\sigma_1(A, B)$ consists of running through all triples of points in $SDR(A)$ and all triples of arcs in $SDR(B)$, to calculate for every combination the rotation R, and to evaluate σ_1 for every valid R. The maximum value is the similarity measure $\sigma_1(A, B)$. Assuming that $SDR(A)$ and $SDR(B)$ have been calculated, this results in the following algorithm outline, called algorithm1.

```
for all points p1    // of SDR(A)
  for all points p2 > p1
    for all points p3 >  p2
      for all arcs a1    // of SDR(B)
        for all arcs a2
          for all arcs a3
            if (tvt(p1, p2, p3, a1, a2, a3, R)){
               sigma1=Vol(A)^{2/3} Vol(B)^{1/3}/Vol(R(A),R(A),B)
               if(sigma1>sigma1_max){sigma1_max=sigma1}
            }
return sigma1_max;
```

In the implementation it is assumed that the arcs and points are stored in a linearly ordered data structure. In this data structure, the variable p1 runs through all points, the variable p2 runs through all points greater than p1, and the variable p3 runs through all points greater than p2. In this way irrelevant permutation evaluations are avoided.

The time complexity of algorithm1 is easily derived. We assume that A and B are approximately of the same complexity, i.e. have approximately the same number of vertices, edges and faces. We denote the number of faces of A and B as f, the number of edges of A and B as e. So, the number of points in $SDR(A)$ equals f, and the number of arcs in $SDR(B)$ equals e. Because e is proportional to f, the inner loop is evaluated $O(f^6)$ times. For polyhedra of small and medium complexity the time consumption of tvt() by far exceeds the timeconsumtion of calculating the mixed volume, so the time complexity of the complete algorithm is $O(f^6)$.

3 Using Geometric Inequalities to Skip Orientations

As explained before, the function tvt() calculates a rotation R with the property that $R(p_1)$ lies on a_1, $R(p_2)$ lies on a_2 and $R(p_3)$ lies on a_3. However, without calling tvt(), it is possible to detect cases where no such R exists. As an example, let us look at two points p_1 and p_2 with a spherical distance $\mathrm{d}(p_1, p_2)$, and at two arcs a_1 and a_2, where $\mathrm{dmin}(a_1, a_2)$ and $\mathrm{dmax}(a_1, a_2)$ are the minimal and maximal distance between the arcs. Here, $\mathrm{dmin}(a_1, a_2)$ is defined as the minimum distance of the points q_1 and q_2 where q_1 lies on a_1 and q_2 lies on a_2, i.e., $\mathrm{dmin}(a_1, a_2) \equiv \{\min(d(q_1, q_2))|q_1 \text{ on } a_1, q_2 \text{ on } a_2\}$. $\mathrm{Dmax}(a_1,a_2)$ is defined analogously. Obviously, only when $\mathrm{dmin}(a_1, a_2) \leq \mathrm{d}(p_1,p_2) \leq \mathrm{dmax}(a_1, a_2)$, p_1 can lie on a_1 while at the same time p_2 lies on a_2, see figure 3. This observation may be used to skip calls of tvt(). Of course, the same principle may be used for the other two pairs of points and arcs, i.e, tvt() should only be called when

$$\mathrm{dmin}(a_1, a_2) \leq d(p_1, p_2) \leq \mathrm{dmax}(a_1, a_2) \text{ and} \tag{7}$$

$$\mathrm{dmin}(a_2, a_3) \leq d(p_2, p_3) \leq \mathrm{dmax}(a_2, a_3) \text{ and} \tag{8}$$

$$\mathrm{dmin}(a_3, a_1) \leq d(p_3, p_1) \leq \mathrm{dmax}(a_3, a_1). \tag{9}$$

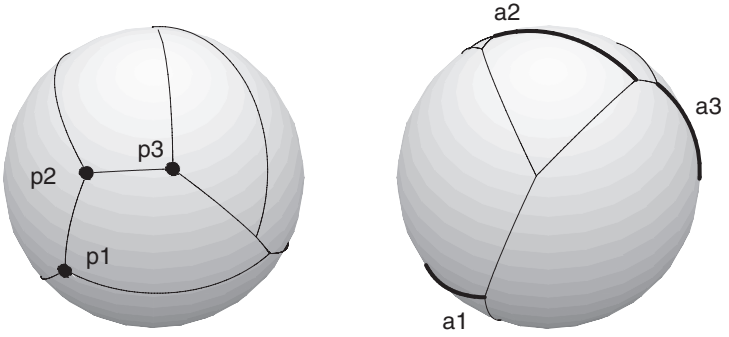


Fig. 3. (a): $SDR(A)$ with three marked points p_1, p_2, p_3. (b): $SDR(B)$ with three marked arcs a_1, a_2, a_3. $SDR(A)$ may be rotated so that in the overlay $R(p_2)$ lies on a_2 and $R(p_3)$ lies on a_3, but clearly then $R(p_1)$ can not lie on a_1.

In the implementation we calculate the distance between all pairs of points of $SDR(A)$ in a preprocessing phase, and store these distances in a table indexed by two points. In the same way we store the minimal and maximal distance between all arcs in $SDR(B)$ in tables indexed by two arcs. Now we can give algorithm2.

```
fill_distance_tables()
for all points p1       // of SDR(A)
  for all points p2 > p1
   for all points p3 > p2
    for all arcs a1      // of SDR(B)
      for all arcs a2
```

```
        for all arcs a3
         if   (dmin(a1, a2)  <= d(p1,p2) <= dmax(a1, a2) and
               dmin(a2, a3)  <= d(p2,p3) <= dmax(a2, a3) and
               dmin(a3, a1)  <= d(p3,p1) <= dmax(a3, a1)){
               if (tvt(p1, p2, p3, a1, a2, a3, R)){
                 sigma1=Vol(A)^{2/3} Vol(B)^{1/3}/Vol(R(A),R(A),B)
                 if(sigma1>sigma1_max){sigma1_max=sigma1}
               }
         }
return sigma1_max;
```

Obviously, the number of calls of tvt() in algorithm2 is less than the number of calls in algorithm1. Moreover, as the complexity of B increases, the arcs in $SDR(B)$ get smaller. The smaller the arcs, the smaller the range of distances between them and thus the smaller the probability that a pair of points will fit between them, resulting in a higher probability that combinations are skipped. I.e. the improvement is not simply a constant factor but is stronger for more complex polyhedra. In the following section we present the experimental time complexity of algorithm1 and algorithm2 , and we derive a first order approximation of the time complexity of algorithm2.

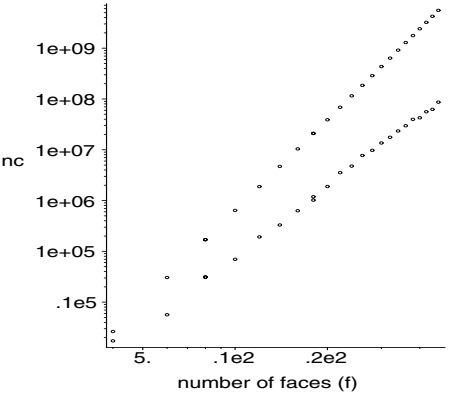


Fig. 4. The results of the experiments with algorithm1 (upper dots) and algorithm2 (lower dots), plotted logarithmically on both axes. In this plot, the results of both algorithms are linear, indicating that the time complexity is of the form $nc = f^e$, where nc is the number of calls of tvt(), f is the number of faces of A and B, and e the exponent. A least square fit in this plot gives $e = 6.0$ for algorithm1, and $e = 4.57$ for algorithm2. So, the experimental time complexity of algorithm1 is $O(f^6)$ and of algorithm2 $O(f^{4.57})$.

4 Results and Complexity Analysis

We tested algorithm1 and algorithm2 on randomly generated polyhedra, ranging in complexity from 4 to 46 faces. The polyhedra A and B were of the same complexity in terms of the number of faces, edges and vertices. To generate polyhedra of the same complexity, for every test we generated a random polyhedron A, and generated random polyhedra until a polyhedron was found with the same number of faces, edges and vertices as A. This polyhedron was assigned to B. For the pair A, B we used algorithm1 and algorithm2 to determine the number of calls of tvt(). In figure 4 the logarithm of the number of calls *nc* of tvt() is plotted as a function of the logarithm of the number of faces. From this plot it can be seen that algorithm2 is significantly faster than algorithm1. For polyhedra with 10 faces algorithm2 is ≈ 10 times faster than algorithm1, and for polyhedra with 46 faces it is ≈ 60 times faster, see figure 5. In the log-log plot, the results of algorithm1 and algorithm2 are both linear, indicating that both algorithms have a time complexity of the form $nc = a.f^e$ where nc is the number of calls of tvt(), f the number faces, and a and e constants. Fitting a line through these points with a least squares method gives for algorithm1 e=6, and for algorithm2 e=4.57. So, the experimental time complexity of algorithm1 is $O(f^6)$ and of algorithm2 $O(f^{4.57})$.

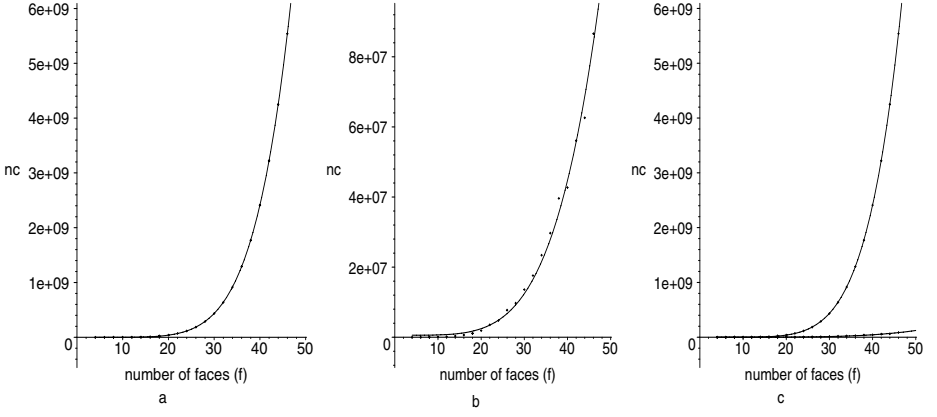


Fig. 5. a: The results of algorithm1. The measured performance is represented by points, and the function f^6 is given as a curve. b: The results of algorithm2. The measured performance is represented by points, and the function $f^{4.57}$ is given as a curve. c: The graphs from a and b in one figure, showing the difference in performance of algorithm1 and algorithm2 on a linear scale.

Now we will derive a first order approximation of the time complexity of algorithm2. The polyhedra used in our experiments are random polyhedra. For these polyhedra it holds that the number of edges is proportional to the number of vertices, and the number of faces is proportional to the number of vertices. In the slope diagram a similar property holds: the number of points, arcs and faces

are proportional to each other. In the derivation we use the easily verified fact that the average arc length in the slope diagram is proportional to $\frac{1}{\sqrt{(f)}}$.

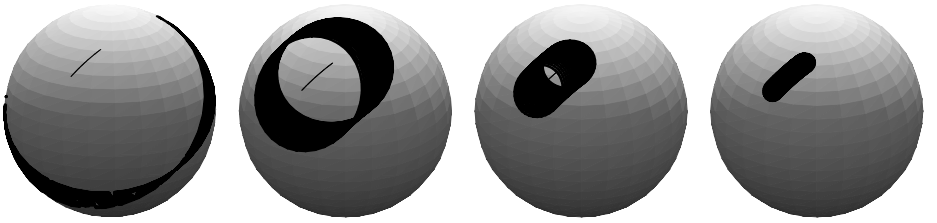


Fig. 6. Four spheres with an arc a1, and regions (black) consisting of all points with a spherical distance d of 1.5, 0.58, 0.25, 0.1 respectively to a1. In the last two figures the arc is (partially) covered by the black area. When a second arc a2 (not shown) has no point in common with the black region then there are no points q1∈ a1 and q2∈ a2 with the property that d(q1, q2)=d. I.e., for two points p1 and p2 with d(p1, p2)=d, the call tvt(p1,p2,..,a1,a2,..) will return false. So, for this situation the call may be skipped.

Let us now look at the following situation. On the unit sphere we draw a small arc $a1$ with length $|a1|$, and we draw the region consisting of all points with a distance d to $a1$. See figure 6. The region consists of a belt with an average width proportional to $|a1|$. So, the area of the belt is proportional to $|a1|$, which is proportional to $\frac{1}{\sqrt{(f)}}$. Now we draw an arc $a2$ on the sphere at a random place. Because A and B are approximately of the same complexity it holds that $|a1| \approx |a2|$. When $a2$ has no point in the black region there are no points $q1 \in a1$ and $q2 \in a2$ with the property that d$(q1, q2)$=d. I.e., for two points p1 and p2 with d(p1, p2)=d, the call tvt(p1,p2,..,a1,a2,..,R) will return false. So, for this situation the call of tvt() may be skipped. The number of arcs in a slope diagram is $\propto f$, and the area of the belt is $\frac{1}{\sqrt{(f)}}$, so, for a given arc $a1$ the number of arcs lying (partially) in the belt is $\propto \sqrt{(f)}$. Therefore, summed over all arcs $a1$, the number of calls of tvt() $\propto f^{1.5}$. More precise, when a1 and a2 run through all arcs of $SDR(B)$, and p1 and p2 run through all points in $SDR(B)$, then the number of times that it holds that $dmin(a1, a2) \leq d(p1, p2) \leq dmax(a1, a2)$ is proportional to $f^{1.5}$. This holds for one pair of points and one pair of arcs. Algorithm2 runs through three pairs of arcs and points, so the the number of calls of tvt() will be proportional to $(f^{1.5})^3 = f^{4.5}$ This agrees reasonably well with our experimental result of $f^{4.57}$. That our experimental result differs slightly from the theoretical result may be caused by the fact that in our experiments we also used some polyhedra with few faces. In figure 4 it can be seen that the slope of the curve of algorithm2 decreases slightly for more complex polyhedra. Leaving out in this figure the first ten data points and fitting a line to the remaining points gives an exponent of 4.52. So, for polyhedra of medium complexity, our experimental complexity corresponds very well with the theoretical complexity.

5 Discussion and Conclusion

The method presented in this article reduces the number of relative orientations to be considered by using geometric distance inequalities. When, for a given set of arcs a_1, a_2, a_3 and points p_1, p_2, p_3, the distance inequalities are not fulfilled then there is no rotation R such that $R(p_1)$ lies on a_1, $R(p_2)$ lies on a_2 and $R(p_3)$ lies on a_3. However, when all three inequalities are fulfilled that does not mean such a rotation exists. That is because, for example, the angle defined by the points $p1, p2, p3$ does not correspond with the range of angle defined by the arcs $a1, a2, a3$. Analogous to the distance inequalities, we can give three angle inequalities. We expect that, by combining distance inequalities with angle inequalities, we can reduce the time complexity of Minkowski-sum based similarity calculations even further.

In this article we presented a method to speed up the search for the relative orientation that minimizes the mixed volume of two convex polyhedra. However, because the volume of the Minkowski sum consists of two mixed volumes (and the orientation independent volumes V(A) and V(B)), the method may also be used to speed up the search for the relative orientation that minimizes the volume of the Minkowski sum. I.e. it may be used for speeding up Minkowski sum based similarity calculations in general.

References

[1] Veltkamp, R.C.: Shape Matching: Similarity Measures and Algorithms. *Shape Modeling International 2001*: 188–196

[2] Heijmans, H.J.A.M. and Tuzikov, A.: Similarity and symmetry measures for convex shapes using Minkowski addition. *IEEE Trans. Patt. Anal. Mach. Intell. 20*, 9 (1998), 980–993.

[3] Tuzikov, A.V., Roerdink, J.B.T.M., and Heijmans, H.J.A.M.: Similarity measures for convex polyhedra based on Minkowski addition. *Pattern Recognition 33*, 6 (2000), 979–995.

[4] Roerdink J.B.T.M. and Bekker H.: Similarity measure computation of convex polyhedra revisited. LNCS vol. 2243, (2001) Springer Verlag.

[5] Bekker, H. and Roerdink, J.B.T.M.: Calculating critical orientations of polyhedra for similarity measure evaluation. In *Proc. 2nd Annual IASTED International Conference on Computer Graphics and Imaging, Palm Springs, California USA, Oct. 25-27* (1999), pp. 106–111.

[6] Sangwine-Yager, J.R.: Mixed volumes. Chapter 1.2 of: *Handbook of convex geometry.* (1993) Eds. Gruber, P.M., Wills, J.M. Elsevier science publishers B.V.

A Practical Algorithm for Approximating Shortest Weighted Path between a Pair of Points on Polyhedral Surface

Sasanka Roy, Sandip Das, and Subhas C. Nandy

Indian Statistical Institute, Kolkata - 700 108, India

Abstract. This paper presents an approximation algorithm for finding minimum cost path between two points on the surface of a weighted polyhedron in 3D. It terminates in finite time. For a restricted class of polyhedron better approximation bound can be obtained.

1 Introduction

Shortest path problems on weighted polyhedral surface in 3D play important roles in geographical information system and robotics. The shortest path problem between two points s and t on the surface of an unweighted polyhedron is studied extensively [1,5,7,10,14,12,15]. The best known algorithm for producing the optimal solution runs in $O(n\log^2 n)$ time [7], where n is the number of vertices of the polyhedron. Two approximation algorithms for this problem were proposed in [15], which can produce paths of length $7(1+\epsilon) \times opt$ and $15(1+\epsilon) \times opt$ respectively, where *opt* is the length of the optimal path between s and t, and ϵ is an user specified degree of precession. The running times are respectively $O(n^{5/3}\log(5n/3))$ and $O(n^{8/5}\log(8n/5))$. For convex polyhedron, an $(1+\epsilon)$-approximation algorithm was proposed in [1], which runs in $O(n/\sqrt{\epsilon})$ time.

In the weighted version of the problem, each face f is attached with a weight $w(f) \in [0, \infty)$. The cost of a path Π between a pair of points s and t on the surface of the polyhedron is defined as follows: *Let Π be the concatenation of a set of line segments $\{\sigma_1, \sigma_2, \ldots \sigma_k\}$, such that each σ_i may be any one of the following two types: (A) it lies completely on a single face f, or (B) it lies on an edge shared by a pair of faces f' and f''. The cost of the path Π, denoted by cost(Π), is equal to $\sum_{i=1}^{k} w_i|\sigma_i|$, where $|\sigma|$ denotes the length of a line segment σ, and $w_i = w(f)$ or $min(w(f'), w(f''))$ depending on whether σ_i is of type A or type B respectively.*

The first work on approximating the minimum cost path on the weighted polyhedral surface appeared in [11]. It uses *continuous Dijkstra* method [10] and exploits the fact that the minimum cost path follows *Snell's law of refraction*. The algorithm locates a path whose cost is within a factor of $(1+\epsilon)$ of that of the optimal path. The running time is $O(n^8 \times \log(\frac{nNW}{\epsilon}))$, where N is the largest integer coordinate among the vertices and W is the maximum integer

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 42–52, 2004.
© Springer-Verlag Berlin Heidelberg 2004

weight among the faces of the polyhedron [13]. An implementable method for this problem appeared in [9], which runs in $\frac{n^3N^2W}{\epsilon w}$ time, where w is the smallest weight among all faces of the polyhedron.

In [8], two algorithms are proposed. The first one adds equally spaced Steiner points on the edges of the polyhedron, and approximates the cost of the optimum path to $(opt + LW)$ in $O(n^5)$ time; here L is the longest edge of the polyhedron. The second one uses graph spanners, and it runs in $O(n^3 \log n)$ time. Here the cost of the reported path is no more than $\beta(opt + LW)$, where $\beta = \frac{1}{cos\theta - sin\theta}$, $\theta = \frac{\pi}{\rho}$ and $\rho > 4$. The best known implementable algorithm for this problem appeared in [2], and it claims to produce a path of weighted length $(1+\epsilon) \times opt$ among a pair of query points in an weighted polyhedron. Several improvements are also proposed [3,4], but there may exist situation(s) where the actual execution of the method proposed in [2,3,4] may be practically impossible. To be more preceise, there exists some pathological instances where the number of steiner points generated by this method may be infinite. So, creating and storing such graph requires infinite time and space.

We propose an alternative scheme of approximating the minimum weight path of polyhedron assuming all its faces to be triangulated. Our algorithm terminates in finite time and the approximation bound is $(1 + \frac{1}{sin\theta})$, where θ is the minimum angle incident at the vertices of all its triangular faces. Further it can be shown that, if $\theta \geq \frac{\pi}{4}$, then the approximation bound is at most 2.

2 Review of Earlier Results

Given a triangulated weighted polyhedron P, two points s and t on its surface, and a given constant $\epsilon \in (0, \frac{1}{2})$, Aleksandrov et. al [2] used the following method for finding an $(1+\epsilon)$ approximation of the minimum cost path from s to t:

Let f and f' be the faces containing s and t respectively. Add s (resp. t) to all the three vertices of f (resp. f'). Thus, the problem reduces to finding the minimum cost path between two vertices in a triangulated polyhedron. Next, Steiner points are added on each edge of P, and an weighted graph G is constructed by putting edge among each pair of Steiner points appearing on two different edges of a face of P, as described below. Finally, Dijkstra's algorithm [6] is used to compute the shortest path in the graph G. We use $dist(a, b)$ to denote the Euclidean distance between two points a and b, and $pdist(a, \mathcal{E})$ to denote the perpendicular distance of a line $\mathcal{E}$ from a point a.

2.1 Construction of Graph G

Let $f_1^v, f_2^v, \ldots, f_k^v$ denote the set of faces adjacent to a vertex v. Define

θ_v = $min_{i=1}^{k}$(angle between two edges of face f_i^v adjacent to v), and
h_v = $min_{i=1}^{k} pdist(v, \mathcal{E}(f_i^v))$, where $\mathcal{E}(f_i^v)$ is the edge opposite to vertex v in face f_i^v.

Let $\delta_v = (1 + \epsilon \times sin(\theta_v))$, if $\theta_v < \frac{\pi}{2}$,
$= (1 + \epsilon)$, otherwise,
and $r_v = \epsilon h_v$.

Steiner points are placed considering each face f adjacent to v separately. Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be the two edges of a face f incident to the vertex v. The Steiner points on $\mathcal{E}_1$ are $p_1, p_2, \ldots p_{\mu_p - 1}$ where $dist(v, p_i) = r_v \times {\delta_v}^{i-1}$, and $\mu_p = \log_{\delta_v} \frac{|\mathcal{E}_1|}{r_v}$. Note that, $dist(p_i, p_{i+1}) = \epsilon \times pdist(p_i, \mathcal{E}_2)$, for $i = 1, \ldots, \mu_p - 2$. The Steiner points $q_1, q_2, \ldots q_{\mu_q - 1}$ on $\mathcal{E}_2$ are generated similarly. This process is performed for each vertex of the polyhedron P. Finally, the graph G is constructed with the generated Steiner points as vertices. Its edges are created as follows:

Consider each face f of P separately. If p and q are two Steiner points on two different sides of f, then put an edge (p, q) with $cost(p, q) = dist(p, q) \times w(f)$. If p and q are two consecutive Steiner points on an edge shared by faces f and f', then also put an edge (p, q) with $cost(p, q) = dist(p, q) \times min(w(f), w(f'))$.

The graph G is the union of the subgraphs for all the triangular faces of the polyhedron P. Next, Dijkstra's algorithm for computing the minimum cost path is applied in the weighted graph G.

2.2 Complexity Analysis

The time complexity of the algorithm proposed in [2] for finding the approximate minimum cost path between a pair of points on the surface of a polyhedron P in $3D$ is $O(nm \log(nm))$, where n is the number of triangular faces of P and m is the number of Steiner points on the longest edge of P. Thus, if L is the length of the longest edge of P, then $m = 2(1 + \log_\delta(|L|/r))$, where $r = min\{r_v \mid$ among all vertices v in $P\}$, $\delta = (1 + \epsilon \times sin(\theta))$, and $\theta = min\{\theta_v \mid$ among all vertices v in $P\}$. On simplification, the time complexity becomes $O(\frac{n}{\epsilon^2} \log n \log \frac{1}{\epsilon})$ (see the table given in [4]). Detailed experiment on this algorithm is performed in [16].

Now consider Fig. 1, where a thin triangle Δuvw is demonstrated with $\theta = \angle uvw$, which is very close to 0. If p_i and p_{i+1} are i-th and $i+1$-th steiner points on uv, then $dist(p_i, p_{i+1}) = \epsilon \times pdist(p_i, vw)$ which is less than $pdist(u, vw) = d$ (say). In Fig. 1, we have demonstrated a set of Steiner points $Q = \{q_1, q_2, \ldots\}$ on the edge uv of Δuvw where $dist(q_i, q_{i+1}) = \epsilon \times d$, for all i. The number of points in set Q is equal to $\frac{dist(u,v)}{\epsilon \times d} = \frac{1}{\epsilon \times sin(\theta)}$. For a given ϵ, this quantity tends to infinity as θ approaches 0. Note that, $d = max_{p_i \in vw} pdist(p_i, vw)$. Thus, the number of points in Q is much smaller than the actual number of Steiner points placed on uv by the algorithm in [2]. This leads to the following remark.

Remark 1. If there exists at least one triangular face in the polyhedron such that one of its angles is very small, then the time and space required for constructing the graph G using the algorithm in [2] may be very high irrespective of whether the optimal path between the query points passes through that face or not.

In [3], the running time of this problem has been improved by a factor of $\frac{1}{\epsilon}$ over that of [2] by removing few edges from graph G using Snell's law of refraction.

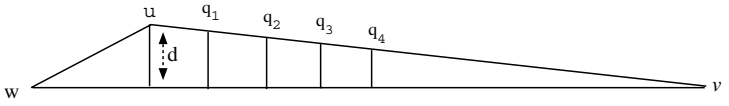


Fig. 1. Justification of the worst case time complexity of the algorithm in [2]

But, as the insertion of Steiner points remains unchanged, this method also suffers from the same problem in such an worst case situation. The last result on this problem appeared in [4], where the time complexity is further improved by another factor of $\frac{1}{\sqrt{\epsilon}}$. But the analysis of running time and the approximation factor is quite surprising. The authors specifically mentioned that the constant inherent in the O-notation includes Γ (average of the reciprocals of the sinuses of the angles of the faces of the polyhedra), and L, which includes the weights of the faces, lengths of the bisectors of the angles in the triangular faces of the polyhedron and several other parameters of the input polyhedron. A careful analysis depicts that there may exist situation where approximation factor becomes drastically worse when the angle between two adjacent faces is close to 0^o or 180^o.

3 Our Proposed Algorithm

We propose an alternative scheme of placing Steiner points on the edges of the polyhedron which guarantees the termination of the algorithm, but the approximation factor depends on the fatness of the triangular faces of the polyhedron.

We consider each edge of P separately and put Steiner points as follows: let uv be an edge and μ be its middle point. We put a Steiner point at μ. Next, we put two sets of Steiner points $p_1, p_2, \ldots p_k$ and $q_1, q_2, \ldots, q_k$ on the segment $u\mu$ and μv respectively, such that p_i (resp. q_i) is the middle point of up_{i-1} (resp. vq_{i-1}), for $i = 1, 2, \ldots, k$, assuming $p_0 = q_0 = \mu$ and $dist(u, p_k) = dist(v, q_k) \leq \epsilon$, for an user-defined constant ϵ (see Fig. 2). We denote the points u and v by p_{k+1} and q_{k+1} respectively. Note that, around each vertex v of the polyhedron, there exists an ϵ-ball such that the portion of each edge (adjacent to v) inside that ϵ-ball contain exactly one Steiner point excepting the vertex v itself. The Steiner points on all the edges are the nodes of the graph G; the edges of G are drawn in the same manner as was done in [2].

Lemma 1. *The number of vertices and edges of the graph G are $O(N \log(\frac{L}{\epsilon}))$ and $O(N(\log(\frac{L}{\epsilon}))^2)$ respectively in the worst case, where N denotes the number of faces of the polyhedron and L is the length of its longest edge.*

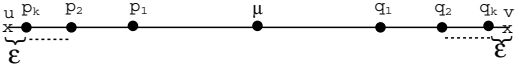


Fig. 2. Our scheme of placing Steiner points

Proof. Follows from the fact that the number of Steiner points on an edge of length λ is equal to $2\log(\lceil\frac{\lambda}{\epsilon}\rceil) - 1$. □

Next, we apply Dijkstra's algorithm on the generated graph as in [2]. Its running time is $O(E)$, where E is the number of edges in the graph G, and it satisfies Lemma 1.

3.1 Analysis of Approximation Factor

While analyzing the approximation factor, we shall restrict ourself to fat triangles, i.e., for each triangular face of the polyhedron, the angle at each vertex is greater than or equal to θ.

Let $a_1, a_2, \ldots, a_k$ be the set of Steiner points on the edge vu of Δuvw when observed from v towards u. Now we have the following results.

Lemma 2. *If a_i and a_{i+1} are two consecutive Steiner points on the edge vu then*
(a) $dist(a_ia_{i+1}) \leq dist(v, a_i)$, and
(b) $dist(a_i, a_{i+1}) \leq \frac{1}{\sin(\theta)} \times pdist(a_i, vw)$.

In order to compute the worst case approximation factor for the cost of the path from s to t along the surface of P using our algorithm, we shall consider an approximated path $\Pi_1(s,t)$ which passes through the same sequence of faces and edges of the polyhedron P as in the optimal path $\Pi(s,t)$. For each line-segment σ on the optimal path, any one of the three cases may arise: (A) it crosses a face, (B) it may pass through a portion of an edge, and (C) it completely lies inside the ϵ-ball attached at a vertex. We approximate σ by σ', where σ' is an edge in G or concatenation of more than one edges in G depending on the aforesaid three cases. The approximation ratio $\frac{|\sigma'|}{|\sigma|}$ is calculated for all $\sigma \in \Pi(s,t)$. An upper bound of the overall approximation factor is the maximum of the approximation ratios considering all the segments on the optimal path $\Pi(s,t)$. It needs to mention that, if the Dijkstra's shortest path algorithm on graph G outputs a different path $\Pi_2(s,t)$, its cost must be less than that of $\Pi_1(s,t)$. Now, we explain the nature of σ' in three different cases.

Case A: When a Segment of the Optimal Path Crosses a Face

Let a line segment of the optimal path crosses a face $f = \Delta uvw$, and it intersects uv and vw at points α and β respectively (see Fig. 3(a)). Let a_1 and a_2 be two Steiner points on the edge uv which appear on two sides of α. Similarly, b_1 and b_2 be two Steiner points on vw which appear on two sides of β. Let a' and b' be the mid-points of a_1a_2 and b_1b_2 respectively. Now, if α is closer to a_i, $i = 1$ or 2, and β is closer to b_j, $j = 1$ or 2, then we approximate the path segment $\alpha \rightarrow \beta$ by $a_i \rightarrow b_j$, whose length is less than or equal to $a_i \rightarrow \alpha \rightarrow \beta \rightarrow b_j$. Thus the approximation factor becomes $\frac{dist(a_i,\alpha)+dist(\alpha,\beta)+dist(\beta,b_j)}{dist(\alpha,\beta)} \leq 1 + \frac{dist(a_i,\alpha)}{dist(\alpha,\beta)} + \frac{dist(\beta,b_j)}{dist(\alpha,\beta)} \leq 1 + \frac{\frac{dist(a_i,a_{i+1})}{2}}{dist(\alpha,\beta)} + \frac{\frac{dist(b_j,b_{j+1})}{2}}{dist(\alpha,\beta)} \leq 1 + \frac{\frac{dist(a_i,a_{i+1})}{2}}{pdist(\alpha,vw)} + \frac{\frac{dist(b_j,b_{j+1})}{2}}{pdist(\beta,uv)} \leq 1 + \frac{1}{\sin(\theta)}$ (from Lemma 2(b)).

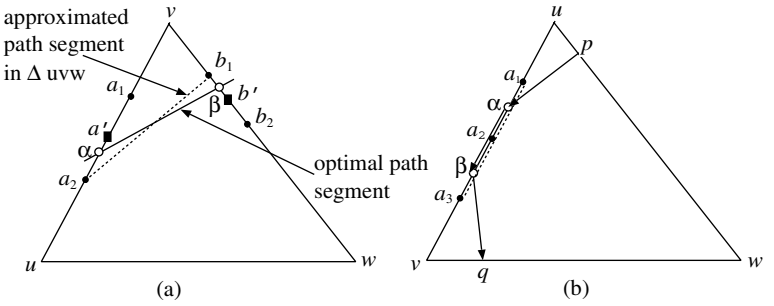


Fig. 3. Calculation of approximation factor inside a face

Case B: When a Segment of the Optimal Path Passes along a Side

Let the optimal path enters in a face $f = \Delta uvw$ through a point $p \in uw$ and exits from f through a point $q \in vw$. But the weights of the other face f' adjacent to the edge uv is small enough such that the optimal path has to pass through uv along a line segment $\alpha\beta$ (see Fig. 3(b)). Let a_1 and a_3 be the Steiner points on uv which are closest to α and β respectively, where a_1, a_2, a_3 are three consecutive Steiner points on uv. We approximate $\alpha\beta$ by a_1a_3, whose length, in this case, may be at most $2 \times dist(\alpha, \beta)$. If the number of Steiner between a_1 and a_3 is more than one, then the approximation factor will surely be less than 2.

Case C: When a Segment of the Optimal Path Passes through an ϵ-Ball

Here we approximate the length of the optimal path by a pair of very small line segments inside the ϵ-ball which are along the sides of the polyhedron and incident at the said vertex. Thus, the total weight will be less than $2\epsilon \times w(f)$.

The analysis of cases A, B and C lead to the following theorem:

Theorem 1. *The length of the path produced by our algorithm is at most*
$$(1 + \tfrac{1}{\sin(\theta)})\Pi(s,t) + 2\epsilon \times (w(f_{\alpha_1}) + w(f_{\alpha_2}) + \ldots + w(f_{\alpha_m})),$$
where $\Pi(s,t)$ is the optimum path from s to t, and $f_{\alpha_1}, f_{\alpha_2}, \ldots, f_{\alpha_m}$ are the faces such that an ϵ-ball of each of these faces contains a complete segment of the optimal path.

Remark 2. Theorem 1 says that, if the optimal path between the query points s and t incidentally passes through a triangular face in the polyhedron having one of its angles very small (i.e., $\theta \to 0$), then our algorithm can not produce good approximation bound on the the length of optimal path. But by Lemma 1, the execution time and space requirement of our algorithm is always finite.

3.2 A More Restricted Model with Better Approximation Bound

We may get better approximation bound if the faces of the polyhedron can be triangulated satisfying the following *nice* property: *each triangular face Δ is*

non-obtuse and the perpendicular distance of each side of Δ from its opposite vertex is less than the length of that side.

Lemma 3. *A triangle satisfying* nice *property, have each angle* $\geq \frac{\pi}{4}$.

The immediate consequence of Lemma 3 is that if the faces of the polyhedron satisfies *nice* property then a trivial upper bound on the weighted length of the path produced by our algorithm is $(1+\sqrt{2})\Pi(s,t) + 2n\epsilon \times W$ (see Theorem 1), where W is the maximum weight among the n faces in P. Below we show that the approximation bound of our algorithm is much better than this trivial bound.

Lemma 4. *If $\alpha\beta$ is a segment of the optimal path passing through the interior of a face Δuvw satisfying* nice *property, and if a and b are the two Steiner points appearing on the same edge of α and β respectively, and are closest to α and β respectively then, $\frac{dist(a,b)}{dist(\alpha,\beta)} < 1.5$.*

Proof. Let us align the side uw of Δuvw with the X-axis. Let $u = (0,0)$, $v = (h,k)$, and $w = (h+\delta, 0)$. Here $h+\delta \geq k$ since Δuvw satisfies the *nice* property.

Let $\alpha\beta$ be the path segment inside $f = \Delta uvw$ and is approximated by ab, where a and b are two Steiner points on uv and vw respectively. In Fig. 4(a), the line segment ab is shown using dotted line, and $\alpha\beta$ is shown using solid line. We prove the lemma by showing $D = 9 \times (dist(\alpha,\beta))^2 - 4 \times (dist(a,b))^2 \geq 0$ considering the following four exhaustive cases :

Case 1 – a Is below the Mid-Point μ of uv and b Is above the Mid-Point γ of vw

This situation is demonstrated in Fig. 4(a). Let us assume that the coordinates of $a = (\frac{h}{2^{i+1}}, \frac{k}{2^{i+1}})$ and that of $b = (h + \frac{\delta}{2^{j+1}}, \frac{2^{j+1}-1}{2^{j+1}}k)$, $i, j \geq 1$. The two neighboring Steiner points of a on uv are $a_1 = (\frac{h}{2^i}, \frac{k}{2^i})$ and $a_2 = (\frac{h}{2^{i+2}}, \frac{k}{2^{i+2}})$, and two neighboring Steiner points of b on vw are $b_1 = (h + \frac{\delta}{2^j}, \frac{2^j-1}{2^j}k)$ and $b_2 = (h + \frac{\delta}{2^{j+2}}, \frac{2^{j+2}-1}{2^{j+2}}k)$. The mid-points of a_1a and aa_2 are $a' = (\frac{3h}{2^{i+2}}, \frac{3k}{2^{i+2}})$ and $a'' = (\frac{3h}{2^{i+3}}, \frac{3k}{2^{i+3}})$ respectively, and mid-points of b_1b and bb_2 are $b' = (h + \frac{3\delta}{2^{j+2}}, \frac{2^{j+2}-3}{2^{j+2}}k)$ and $b'' = (h + \frac{3\delta}{2^{j+3}}, \frac{2^{j+3}-3}{2^{j+3}}k)$ respectively. As ab is the approximation of the optimal path segment $\alpha\beta$ inside Δuvw, α must lie on the line segment $a'a''$ and β must lie on the line segment $b'b''$.

Now, if β is fixed at any point on $b'b''$, the minimum length of $\alpha\beta$ is attained when $\alpha = a'$. Now, in order to prove the lemma, we need to consider the following two subcases:

Case 1.1 – β lies on the line segment bb': Here, assume that the minimum value of $dist(\alpha,\beta)$ is achieved when $dist(b',\beta) : dist(\beta,b) = r : 1$ for some $r \geq 0$. Thus,
$\beta = ((h + \frac{\delta}{2^{j+2}} + \frac{r+2}{r+1} \times \frac{\delta}{2^{j+2}}), (k - \frac{k}{2^{j+2}} - \frac{r+2}{r+1} \times \frac{k}{2^{j+2}}))$, and $dist(\alpha,\beta) = \sqrt{(h - \frac{3h}{2^{i+2}} + \frac{\delta}{2^{j+2}} + \frac{r+2}{r+1} \times \frac{\delta}{2^{j+2}})^2 + (k - \frac{3k}{2^{i+2}} - \frac{k}{2^{j+2}} - \frac{r+2}{r+1} \times \frac{k}{2^{j+2}})^2}$.

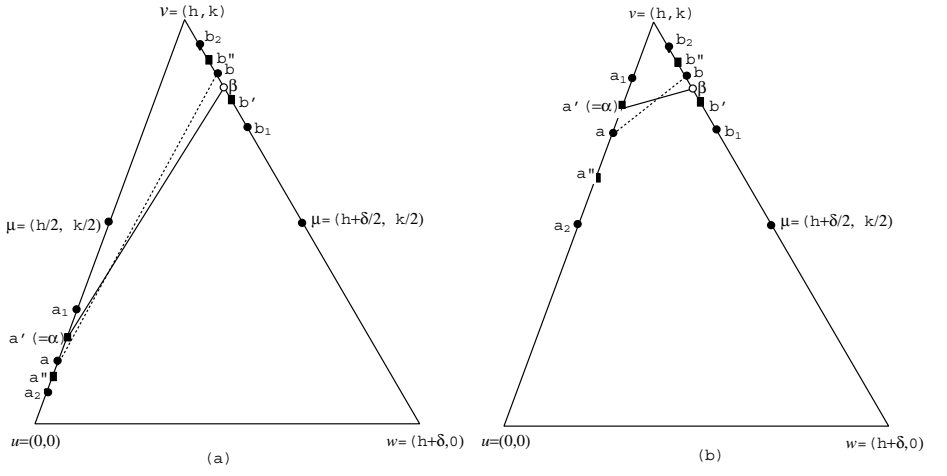


Fig. 4. Proof of Lemma 4

The length of the approximated path
$dist(a,b) = \sqrt{(h - \frac{h}{2^{i+1}} + \frac{\delta}{2^{j+1}})^2 + (k - \frac{k}{2^{i+1}} - \frac{k}{2^{j+1}})^2}$.
By algebraic manipulation $D = 9 \times (dist(\alpha,\beta))^2 - 4 \times (dist(a,b))^2$
$= (5h - \frac{13h}{2^{i+2}} + \frac{10\delta}{2^{j+2}} + \frac{3\delta}{(r+1)2^{j+2}}) \times (h - \frac{5h}{2^{i+2}} + \frac{2\delta}{2^{j+2}} + \frac{3\delta}{(r+1)2^{j+2}}) + (5k - \frac{13k}{2^{i+2}} - \frac{10k}{2^{j+2}} - \frac{3k}{(r+1)2^{j+2}}) \times (k - \frac{5k}{2^{i+2}} - \frac{2k}{2^{j+2}} - \frac{3k}{(r+1)2^{j+2}})$.
If $i \geq 1$ and $j > 1$, $D \geq 0$ for all $r \geq 0$. If $i = j = 1$, $D = \frac{1}{64}((27 \times h + 10 \times \delta + \frac{3\delta}{r+1})(3 \times h + 2 \times \delta + \frac{3\delta}{r+1}) + (k)^2(17 - \frac{54}{r+1} + \frac{9}{(r+1)^2}))$.
Here also, $D > 0$ for all $r \geq 0$, since the value of k can be at most $h + \delta$.

Case 1.2 - β lies on the line segment $b''b$: Here, assume that the minimum value of $dist(\alpha,\beta)$ is achieved when $dist(b'',\beta) : dist(\beta,b) = r : 1$ for some $r \geq 0$. It can be shown that $D > 0$ for all $i, j \geq 1$ and $r \geq 0$.

Case 2 – Both a, b Are above the Line $\mu\gamma$, and ab Is Not Parallel to uw

This case is demonstrated in Fig. 4(b). Without loss of generality assume that $a = (\frac{2^i-1}{2^i}h, \frac{2^i-1}{2^i}k)$ and $b = (h + \frac{\delta}{2^{j+1}}, \frac{2^{j+1}-1}{2^{j+1}}k)$, $1 \leq i \leq j$. Using the same notation as in Case 1, $a_1 = (\frac{2^{i+1}-1}{2^{i+1}}h, \frac{2^{i+1}-1}{2^{i+1}}k)$ and $a_2 = (\frac{2^{i-1}-1}{2^{i-1}}h, \frac{2^{i-1}-1}{2^{i-1}}k)$ are the neighboring Steiner points of a on uv; a' and a'' are the mid-points of aa_1 and aa_2,
$b_1 = (h + \frac{\delta}{2^j}, \frac{2^j-1}{2^j}k)$ and $b_2 = (h + \frac{\delta}{2^{j+2}}, \frac{2^{j+2}-1}{2^{j+2}}k)$ are the adjacent Steiner points of b on vw; b' and b'' are the mid-points of bb_1 and bb_2.

As the optimal path segment $\alpha\beta$ is approximated by ab, $\alpha \in [a', a'']$ and $\beta \in [b', b'']$. Using similar argument as in Case 1, we can say that, for some fixed $\beta \in [b', b'']$, $dist(\alpha,\beta)$ achieves minimum if α coincides with a'. As in Case 1, here also we have considered the two subcases: (i) $\beta \in [b, b']$ and (ii) $\beta \in [b, b'']$, and have observed that the value of $D = 9 \times (dist(\alpha,\beta))^2 - 4 \times (dist(a,b))^2$ is positive for all $j \geq i$.

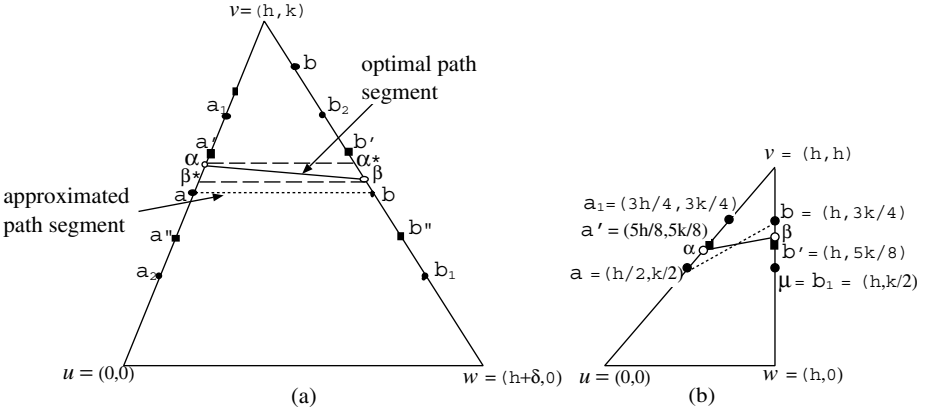


Fig. 5. (a) Illustration of Case 4, (b) An example achieving $\frac{2\sqrt{5}}{3}$ approximation factor

Case 3 – Both the Steiner Points a and b Are below the Line $\mu\gamma$

In this case, D can be shown to be positive in a manner similar to Case 2.

Case 4 – ab Is Parallel to uw

This situation is demonstrated in Fig. 5(a). Here, $a'b'$ is also parallel to uw. Let us draw two line segments parallel to uw from α and β which intersects uv and vw at β^* and α^* respectively. Now, $dist(\alpha,\beta) \geq min(dist(\beta,\beta^*), dist(\alpha,\alpha^*)) \geq dist(a',b')$. Thus, $\frac{dist(a,b)}{dist(\alpha,\beta)} \geq \frac{dist(a,b)}{dist(a',b')} = \frac{dist(u,a)}{dist(u,a')} = C$ (say).

When a is below μ (mid-point of uv), C becomes $\frac{2^{i+2}-2}{2^{i+2}-3}$, assuming $a = (\frac{h}{2^{i+1}}, \frac{k}{2^{i+1}})$ and $b = (h + \frac{2^{i+1}-1}{2^{i+1}} \times \delta, \frac{k}{2^{i+1}})$, and if a is above μ, then D becomes $\frac{4}{3}$. □

Remark 3. It needs to mention that, the approximation factor of $\frac{2\sqrt{5}}{3} \approx 1.4907$ can be achieved for an instance satisfying Case 2 with $i = j = 1$, $\delta = 0$ and $h = k$ (see Fig. 5(b)).

Theorem 2. *The length of the path produced by our algorithm is at most*

$$2 \times \Pi(s,t) + 2\epsilon \times (w(f_{\alpha_1}) + w(f_{\alpha_2}) + \ldots + w(f_{\alpha_m})),$$

where $f_{\alpha_1}, f_{\alpha_2}, \ldots, f_{\alpha_m}$ are the faces such that an ϵ-ball of each of these faces contains a complete segment of the optimal path and is computable in $O(N(log(\frac{L}{\epsilon}))^2)$ time, where N denotes the number of triangulated faces of the polyhedron and L is the length of its longest edge.

Proof. If each segment of the optimal path does not coincide with an edge of the polyhedron, and is not completely contained in the ϵ-ball of a vertex, then by Lemma 4 the approximation factor is bounded above by 1.5. But if there exists instance(s) where the optimum path coincides with edge(s) of the polyhedron then by the analysis of Case B in Section 3.1, the approximation factor can be at most 2. The additive term appears if there exists instances of Case C as described in Section 3.1. □

4 Conclusion

An efficient and implementable algorithm for computing the shortest path between a pair of points on the surface of a weighted polyhedron is proposed. The approximation factor of our method is $1 + \frac{1}{sin\theta}$, where θ is the smallest angle among the triangular faces of the polyhedron. In a restricted case, the approximation factor of our algorithm achieves 2 where each triangular face is non-obtuse, and the perpendicular distance of each side of from its opposite vertex in each of the triangular faces is less than the length of that side. Moreover, if the shortest path is observed to be not passing through any edge of the polyhedron, then the solution can be shown to be $1.5 \times opt$, where *opt* is the length of the shortest path. Though the approximation factor of our algorithm depends on a specific parameter of the input polyhedra, our algorithm outputs good result and always terminates in finite time. We like to mention that, the proof of Lemmata 2, 3, and detailed analysis of the cases in the proof of Lemma 4 are omitted due to the space limitation.

References

1. P. K. Agarwal, S. Har-Peled and M. Karia, *Computing approximate shortest paths on convex polytopes*, Algorithmica, vol. 33, pp. 227–242, 2002.
2. L. Aleksandrov, M. Lanthier, A. Maheshwari and J.-R. Sack, *An ϵ approximation algorithm for weighted shortest paths on polyhedral surfaces*, Proc. Scandinavian Workshop on Algorithmic Theory, LNCS 1432, pp. 11–22, 1998.
3. L. Aleksandrov, A. Maheshwari and J.-R. Sack, *Approximation algorithms for geometric shortest path problems*, Proc. Symp. on Theory of Comput., pp. 286–295, 2000.
4. L. Aleksandrov, A. Maheshwari and J.-R. Sack, *An improved approximation algorithms for computing geometric shortest paths problems*, Proc. Symp. on Foundations of Computing Theory, pp. 246–257, 2003.
5. J. Chen and Y. Han, *Shortest paths on a polyhedron*, Int. J. on Computational Geometry and Applications, vol. 6, pp. 127–144, 1996.
6. E. W. Dijkstra, *A note on two problems in connection with graphs*, Numerical Mathematics, vol. 1, pp. 267–271, 1959.
7. S. Kapoor, *Efficient computation of geodesic shortest paths*, Symp. on Theory of Computing, pp. 770–779, 1999.
8. M. Lanthier, A. Maheswari and J. -R. Sack, *Approximating weighted shortest paths on polyhedral surfaces*, Algorithmica, vol. 30, pp. 527–562, 2001.
9. C. Mata and J. S. B. Mitchell, *A new algorithm for computing shortest paths in weighted planar subdivisions*, Proc. 13th ACM Symp. Comput. Geom. pp. 264–273, 1997.
10. J. S. B. Mitchell, D. M. Mount and C. H. Papadimitrou, *Discrete geodesic problem*, SIAM J. on Computing, vol. 16, pp. 647–668, 1987.
11. J. S. B. Mitchell and C. H. Papadimitrou, *The weighted region problem: finding shortest paths through a weighted planar subdivision*, J. of the Association for Computing Machinary, vol. 38, pp. 18–73, 1991.
12. C. H. Papdimitriou, *An algorithm for shortest path motion in three dimension,* Inform. Process. Lett, vol. 20, pp. 259–263, 1985.

13. J. R. Sack and J. Urrutia, Handbook of computational geometry, North-Holland, Elsevier Science B. V., Netherlands, 2000.
14. M. Sharir and A. Schorr, *On shortest paths in polyhedral space*, SIAM J. Computing, vol. 15, pp. 93–215, 1986.
15. K. R. Varadarajan, P. K. Agarwal, *Approximating shortest path on a non-convex polyhedron*, SIAM J. Computing, vol. 30, pp. 1321–1340, 2000.
16. M. Ziegelmann, *Constrained shortest paths and related problems*, Ph.D. Thesis, Universitat des Saarlandes (Max-Plank Institut fur Informatik), 2001

Plane-Sweep Algorithm of O(nlogn) for the Inclusion Hierarchy among Circles

Deok-Soo Kim[1], Byunghoon Lee[1], Cheol-Hyung Cho[1], and Kokichi Sugihara[2]

[1] Department of Industrial Engineering, Hanyang University,
17 Haengdang-Dong, Sungdong-Ku, Seoul, 133-791, South Korea
dskim@hanyang.ac.kr, {mint,murick}@voronoi.hanyang.ac.kr
[2] Department of Mathematical Informatics, University of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8685, Japan
sugihara@mist.i.u-tokyo.ac.jp

Abstract. Suppose that we have a number of circles in a plane and some of them may contain other circles. Then, finding the hierarchy among circles is of important for various applications such as the simulation of emulsion. In this paper, we present a standard plane-sweep algorithm which can identify the inclusion relationship among the circles in $O(nlogn)$ time in the worst-case. The algorithm uses a number of sweep-lines and a red-black tree for the efficient computation.

1 Introduction

Suppose S is a collection of n circles where a circle C is regarded as a set of points on the circumference. A circle may contain others inside so that there is a hierarchy of inclusion relationships among the circles. In this paper, we assume that circles do not intersect at their circumferences.

Shown in Fig. 1(a) is an example consisting of nine circles and their hierarchical relationships are represented in a tree as shown in Fig. 1(b). For example, the circle C_1 contains two smaller circles C_2 and C_3 inside while the circle C_9 does not contain any other circle. On the other hand, C_4 contains C_6 which again contains two smaller circles C_7 and C_8.

This hierarchical information is represented in the tree shown in Fig. 1(b). For example, C_1, C_4, and C_9 are the top-most circles in the hierarchy and placed at the highest nodes in the tree, except the root node. In this example, C_9 does not contain any other circle and C_1 contains two circles which do not contain any other circle. In the case of C_4, it contains two circles where one of them again contains two other circles.

The first and naive approach to construct the hierarchy among n circles is to compare all pairwise circles, and this approach takes $O(n^2)$ in the worst-case. In this paper, however, we present a standard plane-sweep algorithm which can solve the problem in $O(nlogn)$ time in the worst-case.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 53–61, 2004.
© Springer-Verlag Berlin Heidelberg 2004

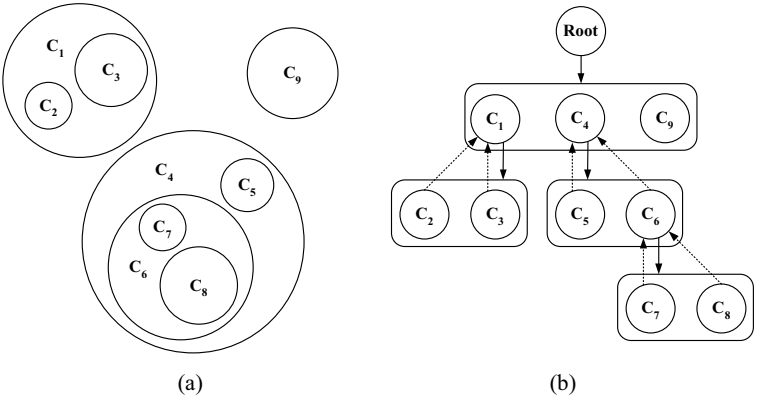


Fig. 1. Nine circles and their inclusion hierarchy shown in a tree. (a) nine circles, (b) the hierarchical relationship shown in a tree among nine circles

2 Motivations

This problem occurs in many applications. Many products in an everyday life such as injection, cosmetics, paint, etc. are emulsions, and different kinds of liquids are intermingled together in an emulsion. In the emulsion, a liquid with less volume exists in the other liquid with larger volume as particles, as shown in Fig. 2(a). It is usual phenomenon that the particles agglomerate as time goes on after an emulsion product is produced [8][9]. Looking at one particle closely usually reveals that the particle itself is in the state of emulsion as well and called a *multiple emulsion*. Fig. 2(b) shows an enlarged picture of one particle and the particle itself contains several other smaller particles. In other words, there are much smaller particles of different material property and the time behavior of this subsystem works independent of the higher level system but works similarly. This kinds of emulsion is usually called a multiple emulsion. Similar phenomenon exists in colloid and ceramics and their time behavior can be similarly analyzed [3].

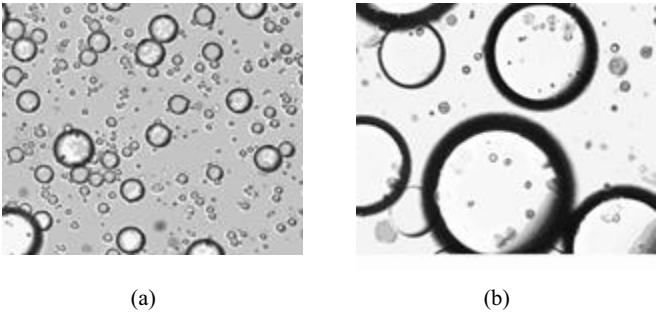


Fig. 2. An example of multiple emulsion (a) higher level, (b) details in a particle

Even though such products are very common in our everyday life, there is not a convenient simulation tool to analyze the time behavior effectively and efficiently. Hence, the developments of these products ordinarily depend on physical experiments and therefore the development cost and time are usually high and long.

To devise an efficient simulation tool for such cases, algorithms and data structures to compute and store the proximity information appropriately are of importance. One answer to this requirement is Voronoi diagrams in general and the Voronoi diagram of circles or spheres in particular for this problem. Voronoi diagram of circles and Voronoi diagram of circles contained in a larger circle have been developed very recently [4][5][6]. As a preprocessing of the simulation tool based on the Voronoi diagram for a multiple emulsion, it is necessary to be able to both recognize such particles in the emulsion and construct their inclusion relationships in a hierarchy. Then, both algorithms of Voronoi diagrams should be incorporated into one since the computation of Voronoi diagrams for both cases should be running simultaneously.

3 Plane Sweeping and Intervals

Suppose that we assign two extreme points for a circle C_i: a leftmost and a rightmost **extreme points** denoted as L_i and R_i, respectively. Hence, there are $2n$ points for n circles. These extreme points are sorted according to their X-coordinate values. Note that their Y-coordinate values are identical to the center of the circle C_i. Let LR-list be an array storing the sorted list of these points.

Among various approaches to solve geometric problems in a plane, the plane-sweep approach provides an efficient algorithm which takes advantage of some kind of orders among the related geometric entities [7]. To apply the idea of plane-sweep algorithm, we consider a moving vertical line, called **the sweep line**, that sweeps the plane from left to right, and define a set of special locations of the sweep line, called **events**, at which the sweep line is tangent to the circles. Among the events, those corresponding to the leftmost extreme points of the circles are called **opening events** since they correspond to events opening new intervals. On the other hand, those for the rightmost extreme points are called **closing events** since some intervals are closed. We call a circle a **generating circle** for an event if it is defined from this circle. For example, the circle C_1 is the generating circle of the event at the extreme points L_1 and R_1. Fig. 3 illustrates an example of three circles, the corresponding opening event, and the related LR-list.

The sweep line at each location is divided into intervals depending on how the sweep line intersects with the circles. Each interval contains two Y-coordinate values, a lower-bound and an upper-bound of the interval, where the interval is effective. We call these values **interval values**. In the given example, the lower and the upper bounds of I_1 are the Y-coordinate value of L_1 and the infinity, respectively. Similarly, those values of I_3 are the negative infinity and the Y-coordinate value of L_1, respectively. In the case of I_2, both values are identical

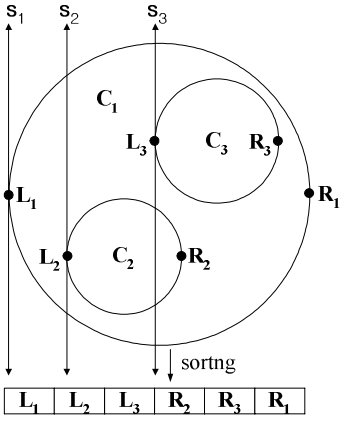


Fig. 3. Line sweeping and interval generation

to the Y-coordinate value of L_1 and therefore I_2 is depicted as a black dot in the figure. We call an interval like I_2 a **zero-interval**.

Initially, we assume that the sweep line is for enough to the left and it consists of only one interval as $[I_1]$ where the interval is defined between the positive and the negative infinities. As the sweep line proceeds to L_1, as shown by S_1, in Fig. 4, two more intervals, I_2 and I_3, are created. Hence, there are three intervals [I_1, I_2, I_3] at S_1. Both the lower bound of I_1 and the upper bound of I_3 are the Y-coordinate value of L_1, and I_2 is a zero-interval.

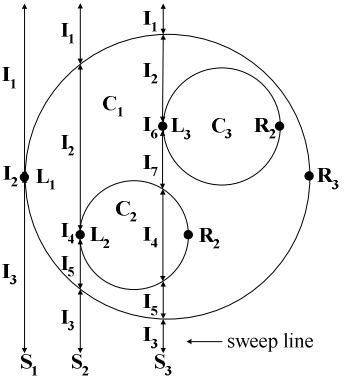


Fig. 4. Line sweeping and interval generation

Then, the sweep line proceeds to the next event represented by S_2, and the interval is further divided into subintervals. The detail is shown in Fig. 5(a), S_2 is tangent to the circle C_2 at L_2 and is always intersecting C_1 at exactly two points since L_2 is between L_1 and R_1. After these intersections are computed, the interval values of I_1, I_2 and I_3 are modified to reflect changed effective intervals due to

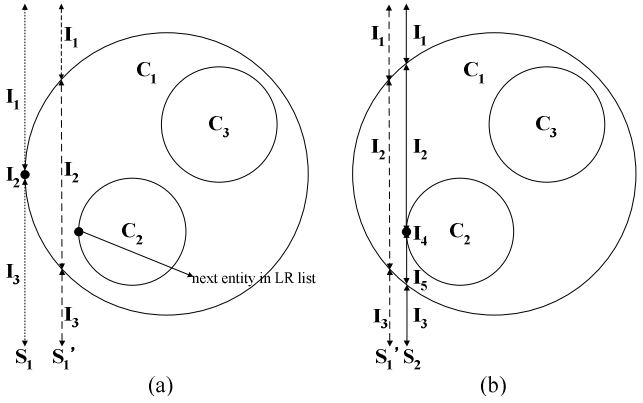


Fig. 5. Transition between two consecutive sweep lines. (a) immediately before hitting new sweep line, (b) immediately after the hitting

the intersection between S_2 and C_1. Note that the effective interval of I_2, which was a zero-interval at S_1, is now between the intersections between C_1 and S_2.

Then, we identify the interval containing L_2 and create two more intervals with appropriate interval values. Note that the interval value of I_2 needs to be updated once more to reflect the creation of new intervals. In Fig. 5(b), for example, L_2 is located in I_2 and therefore the lower-bound of I_2 has to be updated once more with the Y-coordinate value of L_2. Then, new intervals I_4 and I_5 are created with appropriate interval values and inserted below I_2 so that the interval list at S_2 is now $[I_1, I_2, I_4, I_5, I_3]$. In general, therefore, when a new sweep line is met at L_i, we always create two more intervals where one of the intervals is always a zero-interval.

At this point of time, the inclusion relationship between the two circles C_1 and C_2 is determined. If an interval I_k within a circle C_i contains a leftmost extreme point L_j of another circle C_j, the circle C_i includes C_j. In the given example, C_1 includes C_2.

What happens at other opening events lines are similar to what happened at S_2. For example, in the case of S_3 in Fig. 4, the sweep line at S_3 is intersected with both C_1 and C_2, and the result is updated into the list so that C_3 lies in the interval I_2. In opposition to opening events, two intervals are closed at the right most extreme point called closing event. Detail process will be described in section 5.

4 Data Structure for Extreme Points and Intervals

An extreme point, either L_i or R_i corresponding to a circle C_i, consists of an X-coordinate value, a pointer to its generating circle, and a pointer to an interval created as a zero-interval at the extreme point. An extreme point is also pointed from the corresponding generating circle. Note that Y-coordinate value

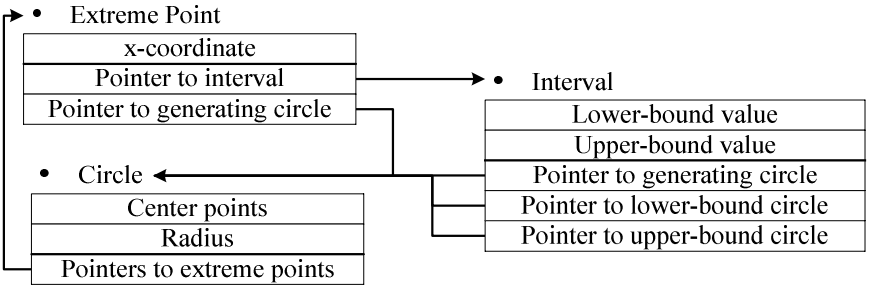


Fig. 6. Data structures of discussed entities

of an extreme point needs not be explicitly stored since it is identical to the Y-coordinate of the center of the generating circle.

An interval consists of a lower-bound value, an upper-bound value, and three different pointers to circles as shown in Fig. 6. There is a pointer to a containing circle in which the interval is included and this pointer is fixed when the interval is created as a zero-interval by the sweep line. When the interval is split by a new leftmost extreme point, this pointer is duplicated to other intervals. In Fig. 5(b), for example, both I_2 and I_5 point to C_1, while I_4 points to C_2 as their containing circles. This pointer, from an interval to its containing circle, is devised to make a decision of the hierarchical relationship between two circles in $O(1)$ time. For example, since L_2 lies in the interval I_2 which points to C_1 as its generating circle, the decision that C_1 contains C_2 can be made by simply looking at the pointer of I_2 at L_2.

In an interval, two more pointers to circles are also needed since an interval consists of a lower-and an upper-bounds where the bounds may be from different circles. These pointers also facilitate $O(1)$ time computation of new values of the interval at different sweep lines.

On the other hand, the intervals are altogether maintained in two different data structures simultaneously: a doubly-linked list and a red-black tree. Intervals are stored in a doubly-linked list in an ascending, or descending, order of the interval values so that the immediate lower and upper interval values can be located in $O(1)$. This is necessary for the constant time update of new interval values. Fig. 6 summarizes the data structures discussed in the above. At the same time, the intervals are also maintained in a red-black tree to facilitate faster operations on intervals, which will be described in the next subsection

5 Intervals in a Red-Black Tree

The intervals created by the line sweeping play the fundamental role in the construction of circle hierarchy. Since the efficient representation of intervals is of importance and the intervals are already ordered according to their Y-coordinate values, the binary search tree is naturally considered as a primary data structure.

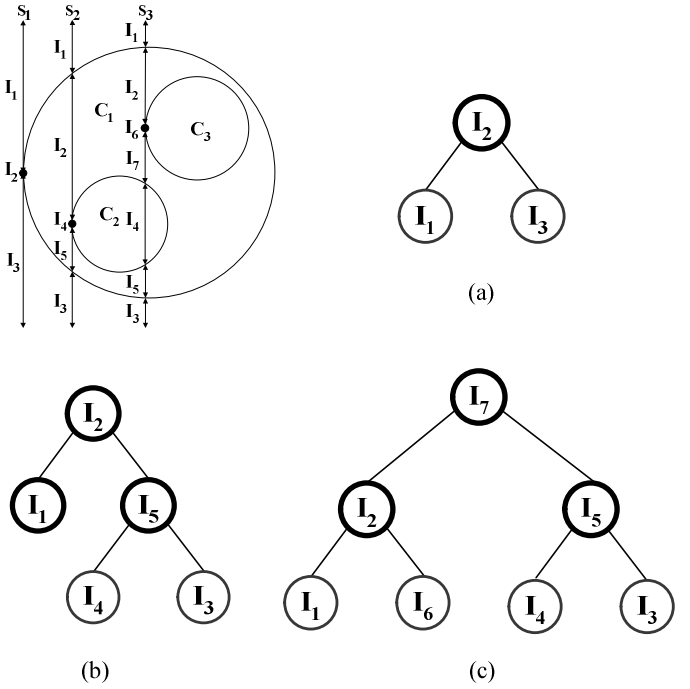


Fig. 7. Examples of red-black trees for intervals at different position of the sweep line. (a) the tree corresponding to S_1, (b) for S_2, and (c) for S_3

Among various binary search trees including AVL and (2,4) trees, we have chosen to employ a red-black tree to store the interval list [1][2]. Even though AVL and (2,4) trees also provide search, insert and delete operations in $O(logn)$ worst-case time complexity like a red-black tree, they have some drawbacks. In the case of the AVL tree, when an insertion occurs, one tri-node restructur-ing operation, called a rotation, is sufficient to restore the height-balance property globally. After an element deletion, however, it is necessary for an AVL-tree to do $O(logn)$ restructuring to restore the height-balance property globally. Similarly, (2,4) tree may need several fusing and split operations to be done after an insertion or dele-tion. On the other hand, a red-black tree, requires only $O(1)$ operation for both inser-tion and deletion to make the tree balanced [2]. Fig. 7 illustrates examples of the tree at different locations of the sweep line.

5.1 Interval Search

Each time the sweep line hits a new circle, a new leftmost extreme point is produced and consequently we need to locate in which interval the extreme point lies.

Since this search decides the hierarchical relationship between two circles, it should be done efficiently. Since the intervals are stored in a red-black tree using the interval values as keys, the appropriate interval can be found in $O(logn)$

time. However, it should be cautioned that, when the sweep line hits a new circle, the interval values at the time of search for a new sweep line, the interval values are old. This means that it is necessary to update the interval values before comparing with current left extreme point.

Since the extreme points are sorted in an ascending order, and it is known that there is no more event other than the event at the current leftmost extreme point, and therefore we traverse from the root of the tree to the bottom while we re-evaluating the new interval values of each node and testing the inclusion of the extreme point in the interval. Since the height of the tree is $O(logn)$, it takes the same amount of time for the search. Note that each interval has pointers to circles producing the upper and lower bounds of the interval. We want to emphasize here that it is sufficient to update the nodes on the search path.

5.2 Interval Insertion

As the sweep line needs a new opening event at S_m, two more intervals, I_k+1 and I_k+2, are created and inserted below the interval I_k containing the corresponding leftmost extreme point L_m. Using the standard insertion operation for a red-black tree, these new intervals can be inserted into the tree in $O(logn)$. After the interval I_k is found in $O(logn)$ in the tree, new intervals are also maintained in the linked list in $O(1)$.

5.3 Interval Deletion

When a rightmost extreme point R_m is selected from the LR-list, the delete operations for appropriate intervals corresponding to L_m have to be executed. Since the closing event at R_m means that the corresponding circle C_m is now being closed, two intervals related to the circle should be removed from the interval data. Note that R_m has a pointer to C_m, and C_m has also a pointer to those intervals.

Hence, those intervals can be deleted from the interval list in $O(1)$ time and can be also deleted from the interval tree using the standard delete operation for a red-black tree in $O(logn)$ time in the worst-case. However, it should be cautioned that the intervals immediately before and after the deleted intervals should be updated in their interval values.

6 Conclusion

In this paper, we have presented a plane-sweep algorithm to construct the hierarchy of inclusion relationship among circles, where circles have different radii and do not intersect at their circumferences. Since there are various applications for this problem in the simulation of physical and chemical processes, the development of an efficient algorithm for this problem is of importance.

Using the plane-sweep method and a red-black tree, we were able to devise an $O(nlogn)$ time algorithm. All of the search, delete and insert operations for appropriate entities in the tree are done in $O(logn)$.

Acknowledgements. The first two authors were supported by the Creative Research Initiative grant from Ministry of Science and Technology in Korea, and the third author was supported by Grant-in-Aid for Scientific Research of the Japanese Ministry of Education, Sports, Culture, Science and Technology.

References

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C.: Introduction to Algorithms, 2nd Edition, The MIT Press (2001)
2. Goodrich, M. T., Tamassia, R.: Data structures and algorithms in Java, 2nd Ed., Wiley (2001)
3. Hong, C.-W.: From Long-Range Interaction to Solid-Body Contact Between Colloidal Surfaces During Forming, Journal of the European Ceramic Society, Vol. 18, (1998) 2159–2167
4. Kim, D.-S., Kim. D., Sugihara, K.: Voronoi Diagram of a circle set from Voronoi Diagram of a point set: I. Topology, Computer Aided Geometric Design, Vol. 18, (2001) 541–562
5. Kim, D.-S., Kim. D., Sugihara, K.: Voronoi Diagram of a circle set from Voronoi Diagram of a point set: II. Geometry, Computer Aided Geometric Design, Vol. 18, (2001) 563–585
6. Kim, D.-S., Kim. D., Sugihara, K.: Voronoi Diagram of Circles in a Large Circle, Lecture Notes in Computer Science, Vol. 2669, (2003) 847–855
7. Nievergelt, J., Preparata, F. P.: Plane-Sweep Algorithms for Intersecting Geometric Figures, Communication ACM, Vol. 25, Issue 10, (1982) 739–747
8. Oh C., Park, J.-H., Shin, S.-I., Oh, S.-G.: O/W/O Multiple Emulsions via One-Step Emulsification Process, Journal of Dispersion Science and Technology, Vol. 25, No. 1, (2004) 53–62
9. Park, J.-H., Oh, C., Shin, S.-I., Moon, S.-K., Oh, S.-G.: Preparation of hollow silica microspheres in W/O emulsions with polymers, Journal of Colloid and Interface Science, Vol. 266, (2003) 107–114

Shortest Paths for Disc Obstacles

Deok-Soo Kim[1], Kwangseok Yu[1], Youngsong Cho[2],
Donguk Kim[1], and Chee Yap[3]

[1] Department of Industrial Engineering
Hanyang University
17 Haengdang-Dong, Sungdong-Ku
Seoul, 133-791, South Korea
dskim@hanyang.ac.kr, {ksyu,donguk}@voronoi.hanyang.ac.kr,
[2] Voronoi Diagram Research Center
Hanyang University, Seoul, South Korea
ycho@voronoi.hanyang.ac.kr
[3] Courant Institute, Department of Computer Science
New York University, New York, U.S.A.
yap@cs.nyu.edu

Abstract. Given a number of obstacles in a plane, the problem of computing a geodesic (or the shortest path) between two points has been studied extensively. However, the case where the obstacles are circular discs has not been explored as much as it deserves. In this paper, we present an algorithm to compute a geodesic among a set of mutually disjoint discs, where the discs can have different radii. We devise two filters, an ellipse filter and a convex hull filter, which can significantly reduce the search space. After filtering, we apply Dijkstra's algorithm to the remaining discs.

Keywords: Geodesic, disc obstacles, ellipse, convex hull, Voronoi diagram, Dijkstra

1 Introduction

Computing geodesics (i.e., shortest paths) and visibility problems for a set of obstacles has been one of the most extensively studied topics in the whole computational geometry [1,2,3,5,6,9,12,13,16,17,18,19]. However, most of the research have focused on polygonal obstacles. We are aware of only a few publications on that treat curved objects [12,13,15,20] and most of them are studied in the point view of mobile robot path planning without serious consideration for computational geometric issues [15,20]. An exception is Pocchiola who provides a nice foundation for the problem in computational geometric point view [12,13].

In this paper, we present an algorithm to compute a geodesic among a number of circular obstacles, called discs, where they have different radii and mutually disjoint. By establishing the characteristics of the geodesic, we have devised two filters, an ellipse filter and a convex hull filter, which reduce the solution space significantly and efficiently. After the solution space is reduced, visibility

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 62–70, 2004.
© Springer-Verlag Berlin Heidelberg 2004

tangent line segments connecting pairs of the unfiltered discs are computed. The the standard Dijkstra algorithm can then be applied to find the shortest path. In this paper, the proposed algorithm has been fully implemented and experimental results are provided. In particular, we demonstract the effectiveness of two proposed filters.

2 Solution Space Reduced by an Ellipse Filter

Suppose S is a collection of pairwise disjoint discs D_i, $i = 1, 2, \ldots, n$. Each $D \in S$ is regarded as an open set, not including its boundary points. A point in $\cup D$ is called an **obstacle point**, and non-obstacle points are said to be **free**.

A path P is a curve with two end points p, q and with a well-defined length, denoted by $d(P)$. If every point in P is free, we call it a **free path**. If there are no other free paths between end points p and q with shorter length than P, then we call P a geodesic (or shortest path). Hence, in our problem, P is a combinatorial path and is a sequence (t_0, A_1, t_1, A_2,..., t_{m-1}, A_m, t_m) where t_i is a tangent line segment between discs and A_i is an arc. Note that t_0 starts from p and t_m ends at q. Note that the upper bound of m is n and points p and q should be also considered in the geodesic in addition to the discs. Let $\mathrm{d}_s(p, q)$ denote the length of a geodesic from p to q, and let $d(p, q)$ denote the length of the line segment $[p, q]$. Then, it is obvious to see that $d(p, q) \leq d_s(p, q)$ since the line segment between two points is the shortest if no disc intersects the line segment.

Lemma 1. *$d_S(p, q) \leq \frac{\pi}{2} d(p, q)$. Moreover, the inequality is an equality iff S contains the disc whose diameter is the segment $[p, q]$.*

Proof. (1) First we claim that $d_S(p, q) \leq (\pi/2)d(p, q)$. To see this, consider the set of discs that intersect the segment $[p, q]$. We construct a path P from p to q which uses all the free points on the segment $[p, q]$. Whenever $[p, q]$ intersects a disc D, we replace the non-free segment

$$[p, q] \cap D$$

by an arc of D. Note that arcs of a disc are always free. There are two arcs determined by the end points of $[p, q] \cap D$, but we always choose the shortest one. It is obvious that the resulting path P satisfies $d(P) \leq (\pi/2)d(p, q)$. Our claim follows since $d_S(p, q) \leq d(P)$.

(2) If S contains the disc with diameter $[p, q]$, then clearly $d_S(p, q) = (\pi/2)d(p, q)$.

(3) It remains to prove that if S does contain this disc, then $d_S(p, q) < (\pi/2)d(p, q)$. Take the path P constructed in part (1). If there are any open subsegment in $[p, q]$ that is free, then clearly $d(P) < (\pi/2)d(p, q)$. Otherwise, it means that there are discs $D_1, \ldots, D_k$ such that $[p, q] \subseteq \cup_{i=1}^{k} \overline{D_i}$. By assumption, $k \geq 2$. Also, if $D_i \cap [p, q]$ is not a diameter of D_i, then $d(P) < (\pi/2)d(p, q)$. So assume that $D_i \cap [p, q]$ is a diameter of D_i for each i. That means that our path P is comprised of half-circles $C_1, \ldots, C_k$ that lie on one side of the line through p, q. Hence $P = C_1; C_2; \cdots; C_k$. Now we produce a shorter path than P by

replacing $C_1; C_2$ by $C_1'; T; C_2'$ where T is that tangent segment from a point p_1 on C_1 to p_2 on C_2, and C_j' are appropriate subarcs of C_j $(j = 1, 2)$. **Q.E.D.**

From the above lemma, we obtain the following corollary:

Corollary 1. *Suppose E is the ellipse with foci p and q such that every point r on E satisfies*

$$\|p - r\| + \|q - r\| = \frac{\pi}{2}\|p - q\|.$$

Then every geodesic from p to q lies within E.

Due to Corollary 1, any geodesic from p to q does not go outside E. Hence, discs outside the ellipse E cannot contribute to a geodesic and thus can be safely removed from the solution space. Since the ellipse provides a convenient mechanism to reduce the solution space, we call it an ellipse filter F_e. Let n_e be the reduced number of discs after an ellipse filter is applied to the original set of n discs.

Filtering extraneous discs using an ellipse filter F_e can be done in $O(n)$ by naively testing all discs whether each disc is outside the ellipse or not. In the average case, however, it can be done more efficiently as follows: After locating the Voronoi regions containing the points p and q, we can traverse neighboring Voronoi regions to check if the disc corresponding to a region is within the ellipse.

Locating a Voronoi region containing a point using the topology of Voronoi diagram of discs can be done by first choosing an arbitrary Voronoi region and then exploring to the neighboring regions while checking if it contains the point. In the exploring process, the idea presented in [4] can be used to efficiently locate region containing the point. Even though this process also takes $O(n)$ in the worst-case, it is known that it takes $O(\sqrt{n})$ on the average[10,11]. If a trapezoidal approach is used, it can be done as efficient as $O(\log n)$ in the worst-case.

Once the Voronoi regions of p and q are known, the discs lying within the ellipse filter can be found in $O(n)$ in the worst-case even if the search is propagated from the Voronoi regions of p and q using the topology information of Voronoi diagram. However, the average time complexity of this search can be as low as $O(n_e)$, where n_e is the number of discs within the filter. Note that the efficient and reliable computation of Voronoi diagram for circles with different radii has been recently known [7,8].

3 Solution Space Reduced by a Convex Hull Filter

In addition to an ellipse filter, there exists another filter called a convex hull filter since it can be proved that a geodesic exists in a convex hull of a few discs around the line segment $\overline{pq}$.

Definition 1. A convex hull of line segment $\overline{pq}$, denoted as $\mathrm{CH}(\overline{pq})$, is the set of points where the convex combination of any two points from the line segment and any discs intersecting the convex hull is also contained in the set.

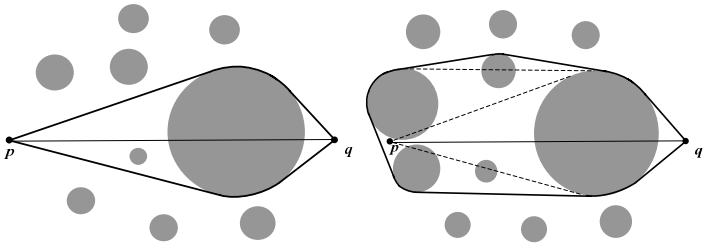


Fig. 1. Definition of convex hull

Hence, the convex hull of a line segment $\overline{pq}$ not intersecting any disc is the segment itself. If the segment intersects discs, then the convex hull is defined around the discs as shown in Fig. 1(a). If the boundary of this convex hull is also intersecting other discs, this new intersecting discs are again included in the definition of convex hull as shown in Fig. 1(b), and this observation yields an algorithm to compute the convex hull in a way similar to shell-growing process Then, the following lemma provides another filter to reduce the solution space.

Lemma 2. *Every geodesic must lie within CH($\overline{pq}$)*

Proof. Suppose that a shortest path P contains a segment exterior to the boundary of CH($\overline{pq}$). Then, P should intersect with the convex hull boundary. Suppose that there are two intersections, x_1 and x_2, between P and the convex hull boundary, without loss of generality. Hence, P consists of three segments divided at these two intersection points. Since the convex hull boundary between x_1 and x_2 is shorter than the segment on P between x_1 and x_2, P cannot be the shortest. This is a contradiction. **Q.E.D.**

The above lemma gives another way to reduce the number of discs. We call it a convex hull filter F_c since it reduces the solution space from n discs to n_c discs. Given a segment $\overline{pq}$, locating all discs intersecting $\overline{pq}$ may take O(n) in the worst-case and O($\sqrt{n}$) on the average by a similar reasoning as explained in the above.

Finding the discs intersecting $\overline{pq}$ starts from the Voronoi region containing p and finds a Voronoi edge of the region which intersects $\overline{pq}$. Then, the disc of which Voronoi region shares this edge can be a candidate of intersecting disc with $\overline{pq}$. However, computing the explicit intersection of $\overline{pq}$ with a Voronoi edge may be quite expensive if the edge is represented in either a rational quadratic Bezier curve or an implicit equation of quadratic order.

Suppose that there are n_i discs on $\overline{pq}$. Then, we sort the discs on $\overline{pq}$ in O(n_i log n_i) in a worst-case. Computing four tangent line segments between all consecutive pairs of discs on $\overline{pq}$ can be computed and stored in O(n_i). Among four tangent line segments for each disc pair, we select two outer tangent line segments. Then, all of the outer tangent line segments are arranged to form a circular chain in a doubly-linked circular list. When this list is created, the tangent line segments are arranged in a counter-clockwise oriented as shown in Fig. 2. Note that the number of edges, n_e, in the *initial edge list* is 2(n_i+1).

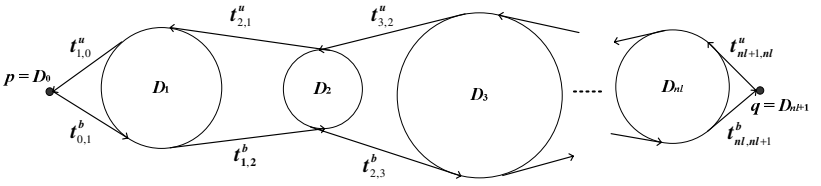


Fig. 2. Circular chain to construct the initial convex hull

Then, a **shell-growing procedure** is applied to find the convex hull as follows. A shell-growing procedure consists of repeated iterations where each iteration is similar to Graham′s scan for a convex hull computation of point set on a plane. Applying the Graham′s scan-like operation to this edge list, we can find the initial convex hull CH($\overline{pq}$)[14]. Ignoring the arcs between consecutive line segments, we check if the consecutive tangent line segments form left or right-turn. If they form a right-turn, both tangent line segments are removed and new tangent line segments are created between two circles. Then this new tangent line segment is tested with the previous tangent line segments, and so on. It can be shown that this Graham′s scan-like process can produce an initial convex-hull from the initial edge list and takes only O(n_e) in the worst-case.

Then, from each boundary edge of the initial convex-hull, we check if this edge intersects other discs or not. After locating all discs on each edge, one of the outer tangent line segments which lies exterior to the current convex-hull, between consecutive pairs of intersecting discs, are inserted in an appropriate places of the edge list. After repeating this process for all boundary edges, we apply the left or right-turn check again. This process terminates when no boundary edge intersects with any disc in the span. Hence, applying this process repeatedly makes the initial convex hull to grow just like a shell grows. The obvious time complexity of the shell-growing procedure is O(n^3) in the worst-case since there can be a linear number of shells, each shell can have a linear number of line segments, and detecting the intersecting discs with boundary of a shell takes a linear time in the worst-case. Note that it takes O($n^2\sqrt{n}$) on the average. However, we believe that a tighter bound can be found.

4 Solution Pool Generation

Suppose that the reduced number of discs is n_f after the initial disc set is filtered by either the ellipse filter or convex hull filter, or both. Since the geodesic consists of tangent line segments and arcs between consecutive tangent line segments, it is necessary to generate all possible tangent line segments between all pairwise discs. Assuming that no two tangent lines are redundant, there can be $4n_f(n_f$ -1)/2 tangent line segments between all possible discs. Since p and q are points from which tangent line segments start, there can be additional $4n_f$ tangent line segments. Hence, there can be $2n_f(4n_f+1)$ tangent line segments in total. Note that the corresponding tangent points on discs are also computed and appropriately stored while the tangents are computed.

Among the tangent line segments, some segments intersecting other discs should be ignored since these segments cannot contribute to a solution. Hence, they are removed from the solution space. Test for the removal of a line segment takes $O(n_f)$ in the worst-case, but takes $O(\sqrt{n_f})$ on the average if it takes advantage of the topology of Voronoi diagram. Hence, the removal test for all edges takes $O(n_f^3)$ in the worst-case but $O(n_f^2\sqrt{n_f})$ on the average since there are $O(n_f^2)$ number of tangent line segments.

Then, the circumferences of discs are also subdivided into a number of arcs at the tangent points on the circumferences. Since no tangent lines are redundant, there can be $4n_f$ tangent lines on a disc: $4(n_f\text{-}1)$ from other discs and 4 from p and q. Note that there can be same number of arcs on the disc. Hence, there can be at most $4n_f^2$ arcs in total. Note that there are also same number of tangent points as well. In total, there are $2n_f(n_f+1)+4n_f^2$ line segments and arcs, and $4n_f^2+2$ tangent points including p and q.

Considering this set as a graph where edges represent the line segments and arcs and vertices represent the tangent points or p and q, we represent this graph in an adjacency list and apply Dijkstra algorithm to this graph to find the shortest path. Then, it takes $O(n_f^2 log\ n_f)$ to find the solution since the number of both vertices and edges are quadratic.

5 Experiments

We have performed experiments of the proposed algorithm with various data sets. The computing environment is as follows: Pentium IV with 2GH clock speed and 512 MB of main memory. Fig. 3 shows the result for 1,000 randomly generated discs. In Fig. 3(a) and (b), the solution space is reduced by only an ellipse filter. In Fig. 3(c) and (d), the identical disc set is reduced by both an ellipse filter and the convex hull filter. As we can see from Fig.3(d), the convex hull filter reduces the space significantly in general. Even though it is possible that the solution space by the convex hull filter only is larger than the solution space by the ellipse filter only depending on the distribution of initial disc set, this case is very rare. Even with 1,000 discs distributed as dense as this example, we were not even able to come up with such a case. In this example, the number of discs after the ellipse filter is 135 and the same number after the convex hull filter is 40. The numbers of valid tangent line segments for after the ellipse filter and the convex hull filter are 2,603 and 640, respectively. The small circles on discs as shown in the enlarged figures are tangent points on the discs. In the figures, there are geodesics, the ellipse and convex hull filters and all pairwise tangent line segments which do not intersect other discs.

Table 1 shows some statistics of the performance of the algorithm in the unit of seconds for 1,000 discs. Column A and B show computation times for the solution spaces reduced by only one filter: the ellipse filter and the convex hull filter, respectively. On the other hand, Column C shows the case that the ellipse filter is first applied and then the convex hull filter is also applied to the reduced set. In Column D, two filters are applied in the reverse order.

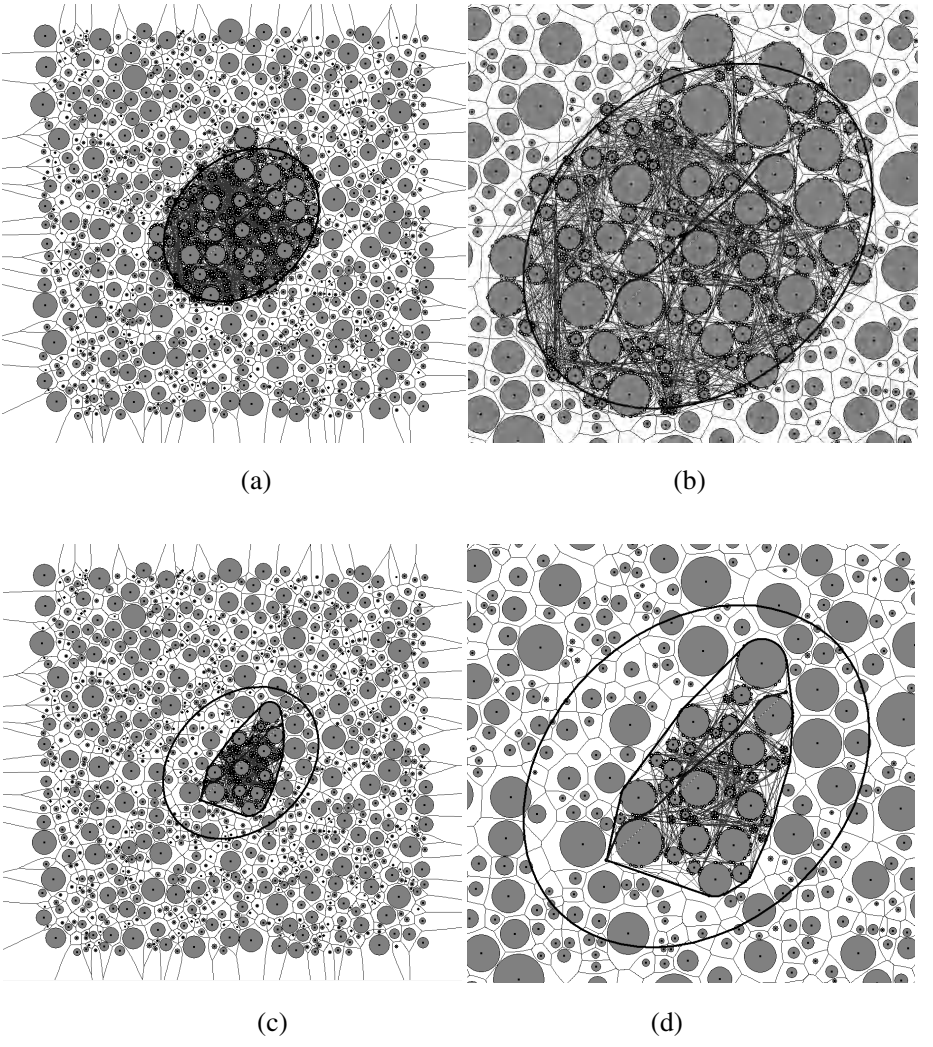


Fig. 3. Results from 1,000 discs. The pairwise tangents are shown in red and the geodesic is shown in blue. (a) and (b) a solution space where the disc set is reduced by an ellipse filter, (c) and (d) the solution space is reduced both an ellipse filter and a convex hull filter.

As shown in the table, the computation time in Column A is significantly larger than the others. It turns out that, in this example, the computation times for Column B and C are similar even though Column B only applies the convex hull filter. This implies that applying the ellipse filter after the convex hull filter does not improve the performance significantly.

Comparing Column C and D reveals rather important point: the ordering of applying the ellipse filter and the convex hull filter is important. Applying an ellipse filter takes $\mathrm{O}(m)$ and it is applied only once for the whole disc set,

Table 1. Computation requirements for different settings in the unit of seconds for 1,000 discs. Column A and B are the time requirements when the solution space is reduced by only one filter: the ellipse filter and the convex hull filter, respectively. Column C is the case when the convex hull filter is applied first and then the ellipse filter is applied. In Column D, both filters are applied in the reverse order.

		A (*Fe* only)	***B*** (*Fc* only)	***C*** (*Fc* and *Fe*)	***D*** (*Fe* and *Fc*)
VD Construction		0.484			
Graph Construction	Filtering	0.000	0.344	0.360	0.016
	Tangent Generation	5.906	0.188	0.187	0.187
Dijkstra		4.875	0.218	0.219	0.219
Sub-total		10.781	0.750	0.766	0.422
Global-total		11.265	1.234	1.250	0.906

Time unit : sec
Number of discs : 1000

where m is the number of discs before the filter is applied. On the other hand, the convex hull filter consists of repeated applications of Graham's scan-like operation and each operation takes $O(m^2)$ in the worst-case. Hence, the total time for convex hull may take $O(m^3)$ in the worst-case. This is the reason why the filtering time for Column C takes much longer than that of Column D.

6 Conclusions

In this paper, we present an algorithm to compute a geodesic among a number of circular obstacles, called discs, where they have different radii and mutually disjoint. By establishing the characteristics of the geodesic, we have devised two filters which can reduce the solution space significantly: An ellipse filter reduces discs which can contribute to the solution by establishing the upper-bound of the geodesic. It turns out that applying an ellipse filter takes only $O(n)$ in the worst-case but reduces the solution space significantly. Then, another filter called a convex hull filter reduces the solution space further by identifying the region where a geodesic can lie. After the solution space is reduces, all possible tangent line segments between discs are computed so that Dijkstra algorithm can be applied to find the shortest path.

In this paper, the algorithm has been implemented and test results for various examples are provided. In particular, we provide the effects of two filters so that they can be efficiently applied.

Acknowledgements. This research was supported by CRI grant from Ministry of Science and Technology, Korea.

References

1. Asano, T., Asano, T., Guibas, L., Hershberger, J., Imai, H.: Visibility of disjoint polyongs, Algorithmica, 1, (1986) 49–63.
2. Alexopoulos, C., Griffin, P.M.: Path planning for a mobile robot,IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22, No. 2, (1992) 318–322.
3. Edelsbrunner, H., Guibas, L. J.: Topologically sweeping an arrangement, J. Comput. System Sci., 38, (1989) 165–194.
4. Gavrilova, M.: On a nearest-neighbor problem under Minkowski and power metrics for large data sets, The Journal of Supercomputing, 22, (2002) 87–98.
5. Ghosh, S.K., Mount, D.: An output sensitive algorithm for computing visibility graphs, SIAM J. Comput., 20, (1991) 888–910.
6. Kapoor, S., Maheswari, S.N.: Efficient algorithms for Euclidean shortest paths and visibility problems with polygonal obstacles, In Proc. 4th Annu. ACM Sympos. Comput. Geometry, (1988) 178–182.
7. Kim, D.-S., Kim, D., Sugihara, K.: Voronoi diagram of a circle set from Voronoi diagram of a point set: I. Topology, Computer Aided Geometric Design, Vol. 18 (2001) 541–562.
8. Kim, D.-S., Kim, D., Sugihara, K.: Voronoi diagram of a circle set from Voronoi diagram of a point set: II. Geometry, Computer Aided Geometric Design, Vol. 18 (2001) 563–585.
9. Mitchell, J.S.B.: Shortest paths among obstacles in the plane, In Proc. 9th Annu. ACM Sympos. Comput. Geom., (1993) 308–317.
10. Mucke, E. P., Saias, I., Zhu, B.: Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations, In Proc. 12th Annu. ACM Sympos. Comput. Geom., (1996) 274–283.
11. Okabe, A., Boots, B., Sugihara, K., Chiu, S. N.: Spatial Tessellations Concepts and Applications of Voronoi Diagrams, John Wiley & Sons, Chichester (2000)
12. Pocchiola, M., Vegter, G.: Computing visibility graphs via pseudo-triangulations, In Proc. 11th Annu. ACM Sympos. Comput. Geom., (1995) 248–257.
13. Pocchiola, M., Vegter, G.: Minimal tangent visibility graphs, Computational Geometry Theory and Applications, Vol. 6, (1996) 303–314.
14. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction, Springer-Verlag, New York (1985)
15. Rimon, E., Koditschek, D.E.: Exact robot navigation using artificial potential functions, IEEE Transactions on robotics and automation, Vol. 8, No. 5, (1992) 501–518.
16. Rohnert, H.: Shortest paths in the plane with convex polygonal obstacles, Inform. Process. Lett., Vol. 23, (1986) 71–76.
17. Rohnert, H.: Time and space efficient algorithms for shortest paths between convex polygons, Inform. Process. Lett., 27, (1988) 175–179.
18. Storer, J.A., Reif, J.H.: Shortest paths in the plane with polygonal obstacles, Journal of the Association for Computing Machinery, Vol. 41, No. 5, (1994) 982–1012.
19. Sudarshan, S., Rangan, C.P.: A fast algorithm for computing sparse visibility graphs, Algorithmica, Vol. 5, (1990) 201–214.
20. Sundar, S., Shiller, Z.: Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation, IEEE Transactions on robotics and automation, Vol. 13, No. 2, (1997) 305–310.

Improving the Global Continuity of the Natural Neighbor Interpolation

Hisamoto Hiyoshi[1] and Kokichi Sugihara[2]

[1] Department of Computer Science, Faculty of Engineering, Gunma University,
1-5-1 Tenjin-cho, Kiryu, Gunma 376-8515, Japan,
hiyoshi@ail.cs.gunma-u.ac.jp,
[2] Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo,
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan,
sugihara@mist.i.u-tokyo.ac.jp

Abstract. The natural neighbor interpolation is a potential interpolation method for multidimensional data. However, only globally C^1 interpolants have been known so far. This paper proposes a globally C^2 interpolant, and write it in an explicit form. When the data are supplied to the interpolant from a third-degree polynomial, the interpolant can reproduce that polynomial exactly. The idea used to derive the interpolant is applicable to obtain a globally C^k interpolant for an arbitrary non-negative integer k. Hence, this paper gets rid of the continuity limitation of the natural neighbor interpolation, and thus leads it to a new research stage.

1 Introduction

Interpolation is an extremely important computational technique in both science and engineering. For example, in order to represent a function occurring in a physical phenomenon on a computer, some discretization process is often required. So we approximate the function using interpolation. Therefore, interpolation has been continuously studied in both the one-dimensional case and the higher-dimensional case.

For the higher-dimensional case, the finite element method (FEM) is widely used as an interpolation method today (There are a lot of textbooks about FEM, e.g., [1]). But there is a potential alternative interpolation method, which is based on the Voronoi diagram, and is called the natural neighbor interpolation. The Voronoi diagram for a site set is the partition of the space by the rule which site is the nearest of all the sites.

In order to see the difference between the two approaches, let us consider a situation where one wants to interpolate a value at a target point from the data given at a finite number of points, called the data sites. With FEM, one first constructs a mesh (e.g., a triangulation) whose vertices are the data sites. Next we find the cell that contains the target point, and sums the data values on the vertices of that cell with the weights that are computed from the locations of the

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 71–80, 2004.
© Springer-Verlag Berlin Heidelberg 2004

target point and the vertices. Note that the number of the data sites appearing in the sum is a constant regardless the configuration of the data sites and the target point.

On the other hand, when using the natural neighbor interpolation, one first computes the Voronoi diagram for the data sites plus the target point. From the obtained Voronoi diagram, one can determine which data sites are near to the target point, in the sense that their Voronoi regions are adjacent to that of the target point. Such data sites are called the natural neighbors. Next one sums the data values given at the natural neighbors with the weights that are computed from the Voronoi diagram. Thus the natural neighbor interpolation takes the configuration of the data sites and the target point into account, varying the number of the data sites used, as the target point moves. For merits of the natural neighbor interpolation, see [2].

In spite of the potential of the natural neighbor interpolation, it is not used widely. One of the reasons is that the continuity of previously known natural neighbor interpolants is limited, while applications sometimes require higher continuity. The first natural neighbor interpolant was proposed about a hundred years ago [5], but it is just a discontinous interpolant. In the early 1980's, the first continous interpolant was proposed by Sibson [2]. This interpolant is continuous, but not globally continuously differentiable. Therefore, some researchers including the authors have been trying to improve the continuity. For the history of the improvement, refer to Sect. 2.

In the authors' previous work [3], we proposed the interpolant that is C^k at the points other than the data sites for any given order k. This paper proposes a method for improving the continuity at the data sites. As a result of these works, we obtain natural neighbor interpolants with globally higher-order continuity. The idea comes from Farin's work [4], in which he combines the natural neighbor interpolation with the theory of Bézier surfaces, and proposed a globally C^1 interpolant. This paper extends Farin's work and construct the natural neighbor interpolation with higher-order global continuity. As a result, we explicitly give a globally C^2 interpolant if we are given not only the values of an underlying function at the data sites but also its gradients and Hessians. The same idea can also be applied to obtain an interpolant with whatever higher continuity.

In this paper, we assume that the dimension of the underlying space is two for the sake of brevity. But this assumption is not essential: the discussion is independent of the dimension.

2 Previous Works

2.1 Sibson's Original Work

First let us see Sibson's interpolant [2] briefly. This paper assumes that the readers are familiar with the Voronoi diagram and its related topics. Refer to, e.g., [6] for the detail.

Assume that we have the data sites $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^2$ and the associated data values $z_1, \ldots, z_n \in \mathbb{R}$, and want to find a function f such that

$$f(\boldsymbol{x}_i) = z_i \quad \text{for } i = 1, \ldots, n. \tag{1}$$

We denote $P = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.

We want to guess the value of the function f at a target point $\boldsymbol{x}$ other than $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Following the idea of the natural neighbor interpolation, we first construct the Voronoi diagram for the point set $P \cup \{\boldsymbol{x}\}$. Assume that $\boldsymbol{x}$ is inside the convex hull of P. Then the Voronoi region of $\boldsymbol{x}$ is bounded. Let A denote the area of the Voronoi region of $\boldsymbol{x}$. The Voronoi region of $\boldsymbol{x}$ can be further decomposed into the regions

$$X_i = \{\boldsymbol{p} \in \mathbb{R}^2 \mid \mathrm{d}(\boldsymbol{p}, \boldsymbol{x}) < \mathrm{d}(\boldsymbol{p}, \boldsymbol{x}_i) < \mathrm{d}(\boldsymbol{p}, \boldsymbol{x}_j), \quad j \neq i\}, \quad i = 1, \ldots, n,$$

where $\mathrm{d}(\boldsymbol{p}, \boldsymbol{q})$ denotes the Euclidean distance between the points $\boldsymbol{p}$ and $\boldsymbol{q}$. Let A^i denote the area of X_i, and $\sigma^i = A^i/A$. In this paper, the superscript simply means an index of a sequence, not a power. Then the following identity holds:

$$\boldsymbol{x} = \sum_{i=1}^{n} \sigma^i \boldsymbol{x}_i. \tag{2}$$

In addition, it follows from the definition that

$$\sum_{i=1}^{n} \sigma^i = 1, \text{ and } \sigma^i \geq 0 \text{ for } i = 1, \ldots, n. \tag{3}$$

Note that in the last inequality, $\sigma^i > 0$ holds only for i such that the Voronoi region of $\boldsymbol{x}_i$ is adjacent to that of $\boldsymbol{x}$. In general, the number of such i's is independent of n when the data sites are uniformly distributed. In this paper, σ^i, $i = 1, \ldots, n$, are called Sibson's coordinates. Note that if the number of the data sites are three, and they are not collinear, then Sibson's coordinates coincide with the barycentric coordinates.

In the above definition, σ^i is not defined over the data site set P. We define σ^i at $\boldsymbol{x}_j$ such that $\sigma^i = \delta^i_j$, where δ^i_j denotes Kronecker's delta. From this definition, (2) holds for $\boldsymbol{x} \in P$ as well.

With σ^i we obtain a natural neighbor interpolant by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \sigma^i z_i. \tag{4}$$

From the definition, (1) follows.

In some context, it is important how high degree of polynomial functions an interpolant can reproduce exactly[1]. In this paper, an interpolant is said to have

[1] In this paper, a polynomial function means a function that can be expressed of a polynomial of cartesian coordinates.

k-th order precision when it can reproduce any degree-k polynomial functions exactly. It can be proved from (2) and (3) that (4) has first-order precision.

Here we briefly state the continuity of Sibson's interpolant. Consider the Delaunay triangulation of P. For each triangle of the triangulation, there is a circle circumscribing the triangle. We call such circles Delaunay circles. Let D denote the set of all the points on the Delaunay circles of P, and let C denote the convex hull of P. Then the following proposition holds:

Proposition 1. *The vector σ^i and hence the interpolant (4) are*

1. C^0 *if* $\boldsymbol{x} \in P$,
2. C^1 *if* $\boldsymbol{x} \in D - P$, *and*
3. C^∞ *otherwise, i.e., if* $\boldsymbol{x} \in C - D$.

Refer to [3] for further explanation. The above proposition says that there are two kinds of points at which (4) is not C^∞.

In the remainder of this section, we review the history of improving the continuity of the natural neighbor interpolation[2].

2.2 Farin's Globally C^1 Interpolant

Farin proposed a globally C^1 interpolant by improving the continuity over P [4]. Before going on, notice that if $f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})$ are C^k, a polynomial of $f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})$ is also (at least) C^k. Farin's idea is to use a polynomial of σ^i's as an interpolant for improving the continuity over P.

Farin's interpolant requires more data than Sibson's; in addition to the values z_i at the data sites, the gradients $\boldsymbol{a}_i$ must be given.

Farin's interpolant is a third-order homogeneous polynomial of σ^i's. Let us introduce a notation for homogeneous polynomials. First, we use Einstein's notation for the sake of simplicity: the expression like $x_i y^i$ actually denotes the sum $x_1 y^1 + \cdots + x_n y^n$. With this notation, a k-th degree homogeneous polynomial can be represented as follows:

$$f = f_{i_1 \ldots i_k} \sigma^{i_1} \cdots \sigma^{i_k}.$$

Because similar terms appears repeatedly in this representation, the coefficients $f_{i_1 \ldots i_k}$ are not determined uniquely. However, if we restrict representations to symmetric ones, the coefficients $f_{i_1 \ldots i_k}$ are uniquely determined. In the following, we represent a k-th degree homogeneous polynomial f as $f = f_{i_1 \ldots i_k} \sigma^{i_1} \cdots \sigma^{i_k}$ with symmetric coefficients $f_{i_1 \ldots i_k}$. One of the merits using this notation instead of the Bernstein-Bézier form (see, e.g., [7]) is simplicity when the differentiation rule is given. When we write the partial differentiation with respect to σ^i as ∂_i, the rule is given as follows:

$$\partial_j f = k f_{i_1 \ldots i_{k-1} j} \sigma^{i_1} \cdots \sigma^{i_{k-1}}.$$

[2] Although Sibson [2] proposed a globally C^1 interpolant as well, it is not listed here. The reason is that it seems to be rather ad hoc, and we could not have used it for improving the continuity further.

From the given data, we define the following quantity:

$$z_{i,j} = \boldsymbol{a}_i \cdot \overrightarrow{\boldsymbol{x}_i \boldsymbol{x}_j}.$$

Then Farin's interpolant is represented as follows:

$$f(\boldsymbol{x}) = f_{ijk}\sigma^i\sigma^j\sigma^k. \tag{5}$$

Here the coefficients are determined as follows:

$$\begin{aligned} f_{iii} &= z_i, \\ f_{iij} &= z_i + \frac{1}{3}z_{i,j}, \\ f_{ijk} &= \frac{z_i + z_j + z_k}{3} + \frac{z_{i,j} + z_{i,k} + z_{j,i} + z_{j,k} + z_{k,i} + z_{k,j}}{12}, \end{aligned}$$

where i, j and k are different from each another. For efficiency reason, we should compute only the coefficients f_{ijk} such that all σ^i, σ^j and σ^k are non-zero.

The next proposition states properties of Farin's interpolant briefly:

Proposition 2. *1. The interpolant (5) is*
a) C^1 *if* $\boldsymbol{x} \in D$, *and*
b) C^∞ *otherwise, i.e., if* $\boldsymbol{x} \in C - D$.
2. The interpolant (5) has second-order precision.

2.3 The Authors' Previous Work

The authors' previous work [3] introduced the following concept:

Definition 1. *Assume that the point set* $P = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^2$ *are given. Let* C *denote the convex hull of* P. *If the functions* $s^i(\boldsymbol{x}) : C \to \mathbb{R}$, $i = 1, \ldots, n$, *have the following properties, then* $(s^1, \ldots, s^n)$ *are called generalized barycentric coordinates: for any* $\boldsymbol{x} \in C$,

$$\boldsymbol{x} = \sum_{i=1}^{n} s^i \boldsymbol{x}_i, \quad \sum_{i=1}^{n} s^i = 1, \text{ and } \sigma^i \geq 0 \text{ for } i = 1, \ldots, n.$$

Note that in the above definition, if $n = 3$, and $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are not collinear, then generalized barycentric coordinates coincide with the barycentric coordinates. From (2) and (3), Sibson's coordinates are generalized barycentric coordinates.

The authors showed that for an arbitrary given non-negative integer k, there exists generalized barycentric coordinates s^i that are C^k over $C - P$. With s^i, we can construct the following interpolant:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} s^i z_i. \tag{6}$$

The next proposition states properties of the interpolant (6) briefly:

Proposition 3. *1. The interpolant (6) is*
a) C^0 *if* $\boldsymbol{x} \in P$, *and*
b) C^k *if* $\boldsymbol{x} \in D - P$, *and*
c) C^∞ *otherwise, i.e., if* $\boldsymbol{x} \in C - D$.
2. The interpolant (6) has first-order precision.

3 Interpolants with Higher-Order Continuity

3.1 Globally C^2 Interpolant

As described in the last section, Farin's technique improves the continuity over P, while the technique proposed in the authors' previous work over $D - P$. The idea for obtaining interpolants with globally higher-order continuity is combining these techniques. Since we have already achieved C^k continuity over $D - P$, we only have to improve continuity over P. Because there is not sufficient space for giving full explanation, only a rough sketch is described.

In Farin's interpolant, the coefficients of the monomials $\sigma^i\sigma^i\sigma^i$ and $\sigma^i\sigma^i\sigma^j$ are uniquely determined so that the interpolant is globally C^1 and can reproduce the data at the data sites. On the other hand, the coefficients of the monomials $\sigma^i\sigma^j\sigma^k$, $i \neq j, k$, cannot be determined in this way. In fact, the value and the gradient of the monomial $\sigma^i\sigma^j\sigma^k$, $i \neq j, k$, at $\boldsymbol{x}_i$ are always zero. In other words, we could choose the coefficients of such monomials arbitrarily without affecting the values and gradients of the interpolant at $\boldsymbol{x}_i$'s. However, we should not do so because doing so makes the interpolant as a whole extremely bumpy. For avoiding such an effect, the coefficients were determined so that the interpolant has second-order precision. However, the coefficients are not unique from this restriction, as was pointed out in [4].

This discussion can be extended for obtaining globally C^k interpolant. In the following, we only give a globally C^2 interpolant explicitly.

For the proposed interpolant, we require more data than Farin's; we assume that in addition to the values z_i and the gradients $\boldsymbol{a}_i$ at the data sites, the Hessians B_i are given.

From the given data, we define the following quantities:

$$z_{i,j} = \boldsymbol{a}_i \cdot \overrightarrow{\boldsymbol{x}_i\boldsymbol{x}_j}, \quad z_{i,jk} = \overrightarrow{\boldsymbol{x}_i\boldsymbol{x}_j} \cdot (B_i\overrightarrow{\boldsymbol{x}_i\boldsymbol{x}_k}).$$

Then the proposed interpolant is expressed as follows:

$$f(\boldsymbol{x}) = f_{ijklm}s^i s^j s^k s^l s^m. \tag{7}$$

Here the coefficients are determined as follows:

$$f_{iiiii} = z_i,$$
$$f_{iiiij} = z_i + \frac{z_{i,j}}{5},$$

$$
\begin{aligned}
f_{iiijj} &= z_i + \frac{2z_{i,j}}{5} + \frac{z_{i,jj}}{20},\\
f_{iiijk} &= z_i + \frac{z_{i,j} + z_{i,k}}{5} + \frac{z_{i,jk}}{20},\\
f_{iijjk} &= \frac{z_i + z_j}{2} + \frac{3(z_{i,j} + z_{j,i})}{20} + \frac{z_{i,k} + z_{j,k}}{10} + \frac{z_{i,jk} + z_{j,ik}}{30} + \frac{z_{i,jj} + z_{j,ii}}{120},\\
f_{iijkl} &= \frac{7z_i + z_j + z_k + z_l}{10} + \frac{11(z_{i,j} + z_{i,k} + z_{i,l})}{90} + \frac{z_{i,jk} + z_{i,jl} + z_{i,kl}}{45}\\
&\quad + \frac{z_{j,i} + z_{j,k} + z_{j,l} + z_{k,i} + z_{k,j} + z_{k,l} + z_{l,i} + z_{l,j} + z_{l,k}}{45}\\
&\quad + \frac{z_{j,ik} + z_{j,il} + z_{j,kl} + z_{k,ij} + z_{k,il} + z_{k,jl} + z_{l,ij} + z_{l,ik} + z_{l,jk}}{180},\\
f_{ijklm} &= \frac{z_i + z_j + z_k + z_l + z_m}{5} + \frac{z_{i,j} + z_{i,k} + z_{i,l} + z_{i,m} + \cdots}{30}\\
&\quad + \frac{z_{i,jk} + z_{i,jl} + z_{i,jm} + z_{i,kl} + z_{i,km} + z_{i,lm} + \cdots}{180},
\end{aligned}
$$

where i, j, k, l and m are mutually different. For efficiency reason, we should compute only the coefficients f_{ijklm} such that all s^i, s^j, s^k, s^l and s^m are non-zero.

The coefficients f_{iiiii}, f_{iiiij}, f_{iiijj} and f_{iiijk} are determined uniquely so that the interpolant is globally C^2 and can reproduce the data at the data sites. The other coefficients, i.e., f_{iijjk}, f_{iijkl} and f_{ijklm}, were determined so that the interpolant has third-order precision, although they are not unique.

Now let us compare the above interpolant with an FEM interpolant, called Q_{18} [8], that can be applied in the same problem setting. For Q_{18}, the space is decomposed into a triangular mesh. As the data, the function values, gradients, and Hessians are given at the vertices of the mesh. The interpolant Q_{18} is a fifth-degree polynomial of the barycentric coordinates when it is restricted to each triangle. The interpolant Q_{18} is globally C^1, and has fourth-order precision. So the proposed interpolant has higher continuity but lower precision than Q_{18}.

Table 1 summarizes the continuity and precision of the interpolants that are described in this paper.

Table 1. The continuity and precision of natural neighbor interpolants

Interpolant	Over P	Over $D - P$	Precision
Sibson's interpolant (4)	C^0	C^1	first order
Farin's interpolant (5)	C^1	C^1	second order
Authors' previous interpolant (6)	C^0	C^k	first order
Proposed interpolant (7)	C^2	C^k	third order
Q_{18}	(globally C^1)		fourth order

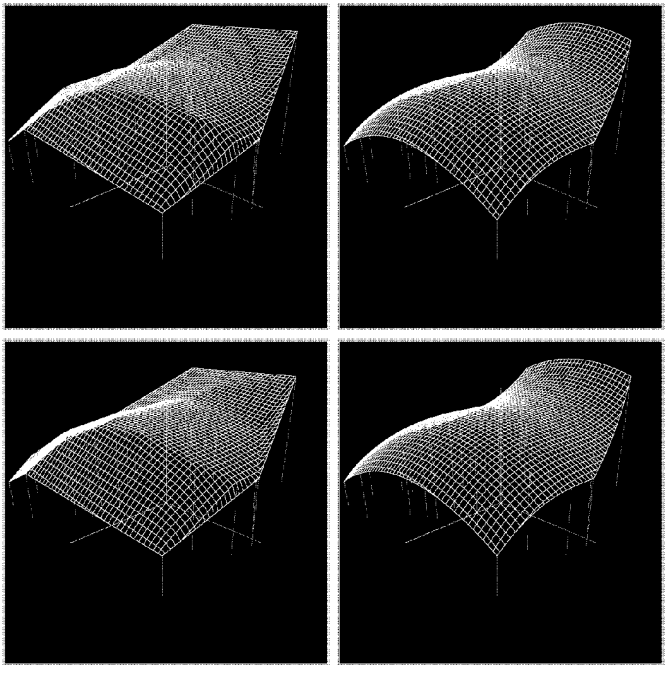


Fig. 1. Example of surfaces created by Sibson's interpolants (upper, left), Farin's interpolant (upper, right), the interpolant proposed in the authors' previous work (lower, left), and the proposed interpolant (lower, right). The number of the data sites is thirty, and the data values are computed from the function $(5/2 - x^3 - y^4)/3$.

3.2 Globally C^k Interpolant

Here we briefly describe how to obtain interpolants with higher-order continuity. Assume that we want a globally C^k interpolant. In this case, we require the partial differential coefficients of up to the k-th order at the data sites. The interpolant is a $(2k+1)$-th degree homogeneous polynomial of generalized barycentric coordinates s^i that is C^k over $C - P$, for example, the k-th order standard coordinates given in [3]. The coefficients of the monomials with $(k + 1)$ or more s^i are uniquely determined from the given data. The remaining coefficients can be determined so that the interpolant has $(k + 1)$-order precision, although they are not unique. This is our general strategy for constructing the interpolant with higher-order continuity. Applying this strategy for individual k, we shall obtain the associated interpolant explicitly.

4 Experiments

We implemented the proposed interpolant with the standard coordinates given in [3] using Java with Java 3D API. The figures given in this paper were created by this program.

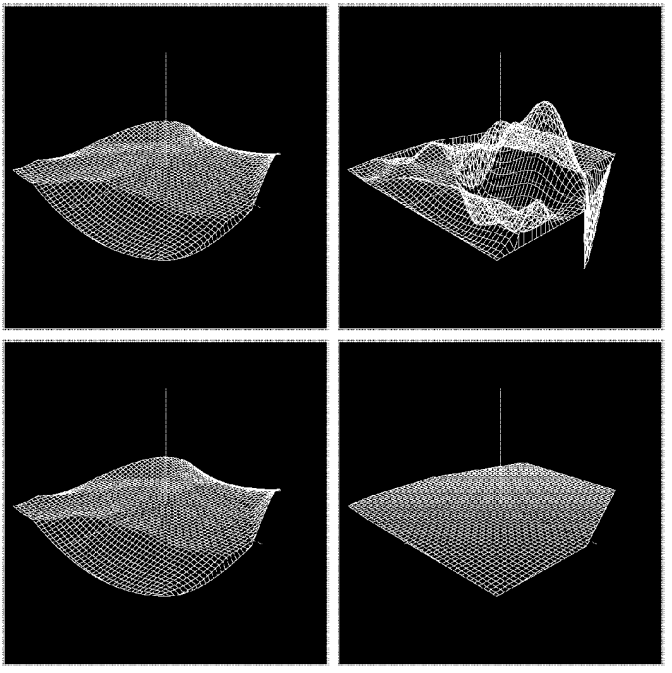


Fig. 2. Errors of Sibson's interpolants (upper, left), Farin's interpolant (upper, right), the interpolant proposed in the authors' previous work (lower, left, the z direction was multiplied by scale factor 10^2), and the proposed interpolant (lower, right, the z direction was multiplied by scale factor 10^6). The number of the data sites is thirty, and the data values are computed from the function $(5/2 - x^3 - y^2)/3$.

Figure 1 shows the line drawings of sample surfaces created by the interpolants described in the paper. As generalized barycentric coordinates, Sibson's coordinates were used for Sibson's interpolant and Farin's interpolant, and the second-order standard coordinates [3] were used for the interpolants (6) and (7). The number of the data sites is thirty, and the data sites were chosen randomly in the region $[-1, 1] \times [-1, 1]$. The data were computed from the function $(5/2 - x^3 - y^4)/3$. In the figure, the data values are represented by the vertical line segments, whose lower endpoints are on the plane $z = 0$. In the figure, we can see sharp apices in the surfaces obtained from (4) and (6) at the data sites, while the other two are smooth.

Next we examine precision of the interpolants. Figure 2 shows errors of the interpolants for the data sampled from the function $(5/2 - x^3 - y^2)/3$, that is a third-degree polynomial. In the figure, the errors were plotted in the z direction. In particular, the errors in Farin's interpolant and the proposed interpolant are shown after they were enlarged by multiplying the scale factor 10^2 and 10^6, respectively. Because the proposed interpolant has third-order precision, we observe that it can reproduce the original function exactly. On the other hand, the other three cannot reproduce the original function exactly, although Farin's

interpolant can reproduce it better than the other two, because Farin's has much better precision.

5 Concluding Remarks

This paper gave an explicit expression of a globally C^2 interpolant based on Voronoi diagrams. It is a fifth-degree homogeneous polynomial of generalized barycentric coordinates; it can reproduce the given data, and it has third-order precision. In general, for any given non-negative integer k, it is possible to construct a globally C^k interpolant with $(k+1)$-th precision.

One of future works will be to find the law which governs the coefficients of the interpolant for an arbitrary k, and to give the interpolants explicitly.

Another will be to develop a general framework for applications of the natural neighbor interpolation. In this paper, we used the term "FEM" rather narrowly so as simply to mean a interpolation method, but it actually is a general framework for solving partial differential equations in science and engineering. The natural neighbor interpolation might be applied for a similar purpose; we want to consider possible application in this direction, i.e., application something like a meshless method for solving partial differential equations.

Acknowledgement. This work is supported by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports, Culture and Technology of Japan.

References

1. Strang, G., Fix, G.J.: An Analysis of the Finite Element Method. Prentice-Hall (1973)
2. Sibson, R.: A brief description of natural neighbour interpolation. In Barnett, V., ed.: Interpreting Multivariate Data. John Wiley & Sons (1981) 21–36
3. Hiyoshi, H., Sugihara, K.: Improving continuity of Voronoi-based interpolation over Delaunay spheres. Computational Geometry: Theory and Applications **22** (2002) 167–183
4. Farin, G.: Surfaces over Dirichlet tessellations. Computer Aided Geometric Design **7** (1990) 281–292
5. Thiessen, A.H.: Precipitation averages for large areas. Monthly Weather Review **39** (1911) 1082–1084
6. Preparata, F.P., Shamos, M.I.: Computational Geometry. Springer-Verlag (1985)
7. de Boor, C.: B-form basics. In Farin, G., ed.: Geometric Modeling: Algorithms and New Trends. SIAM (1987) 131–148
8. Barnhill, R.E., Farin, G.: C^1 quintic interpolation over triangles: two explicit representations. International Journal for Numerical Methods in Engineering **17** (1981) 1763–1778

Combinatories and Triangulations*

Tomas Hlavaty** and Václav Skala***

University of West Bohemia,
Department of Computer Science and Engineering,
Univerzitni 8, 306 14 Plzen,
Czech Republic
{thlavaty,skala}@kiv.zcu.cz

Abstract. The problem searching for an optimal triangulation with required properties (in a plane) is solved in this paper. Existing approaches are shortly introduced here and, specially, this paper is dedicated to the brute force methods. Several new brute force methods that solve the problem from different points of view are described here. Although they have NP time complexity, we accelerate the time needed for computation maximally to get results of as large sets of points as possible. Note that our goal is to design the method that can be used for arbitrary criterion without another prerequisite. Therefore, it can serve as a generator of optimal triangulations. For example, those results can be used in verification of developed heuristic methods or in other problems where accurate results are needed and no methods for required criterion have been developed yet.

1 Introduction

Assume that N points (in a plane) are given. Construct a triangulation on this set of points that is optimal from the point of view of required properties.

The mentioned problem above try to solve many applications and criterions that describe the properties of triangulations can have many forms (e.g., a triangulation that minimizes sum of edge weights or that maximizes minimal angle in triangles, etc.). This paper is just dedicated to this issue and several algorithms that solve this problem are described here.

Next two chapters are a short introduction about triangulations and approaches of triangulation generating. The first chapter is dedicated to the definition of triangulation and to the general properties of triangulations. The second one contains an overview of existing approaches that can solve this issue. The remainder chapters are dedicated to methods based on the brute-force approach and they describe several algorithms. The paper is finished by a comparison of the individual methods mutually and by a conclusion. Note that the comparison is based on implementation of methods for a given problem, exactly, they search for the MWT (i.e., Minimum Weight Triangulation) [5], [7], [10].

* This work is supported by the Ministry of Education of the Czech Republic projects:
** FRVS 1342/2004/G1, *** MSM 235200005.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 81–89, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Triangulation

First of all, we should define the term *triangulation*. However, no exact definition exists. The triangulation can be seen from several views as it is shown in following two definitions (we only will think about triangulation of points in a plane here):

Definition 1. Let us assume that we have a set of different points in a plane $S = \{p_i\}$, $p_i \in E^2$, $i = 1, \ldots, N$. Then a set of so called edges represents a triangulation $T(S) = \{e_i\}$ if the following conditions are valid:

1. Each edge e_i in the triangulation includes just two points from the set S and these points are end points of the edge (the edge is an abscissa that connects two given end points).
2. Two arbitrary edges from the triangulation do not cross mutually.
3. It is impossible to insert another edge into the triangulation and to keep the previous conditions valid simultaneously.

Definition 2. Let us assume that we have a set of different points in a plane $S = \{p_i\}$, $p_i \in E^2$, $i = 1, \ldots, N$. Then a set of so called empty triangles represents a triangulation $T(S) = \{t_i\}$ if the following conditions are valid:

1. Each triangle in the triangulation includes just three points from the set S and these points are vertices of the triangle (another point inside the triangle cannot be included - this triangle is called the *empty triangle*).
2. Intersection of two arbitrary empty triangles from the triangulation can be a vertex or an edge of the triangle maximally.
3. It is impossible to insert another empty triangle into the triangulation and to keep the previous conditions valid simultaneously.

In the first moment, the definitions seem to be similar. It is valid because they only look on the triangulation from two different views. In the first definition the triangulation is represented as a set of edges and in the second one the triangulation is represented as a set of triangles. An example of a triangulation is shown on the Fig. 1. Note that many other definitions can be made up.

The boundary of the triangulation is the convex hull of a set of points S (see the Fig. 1). Note that this is always valid for all triangulations constructed according to the mentioned definitions and we can use this fact to determine those edges automatically.

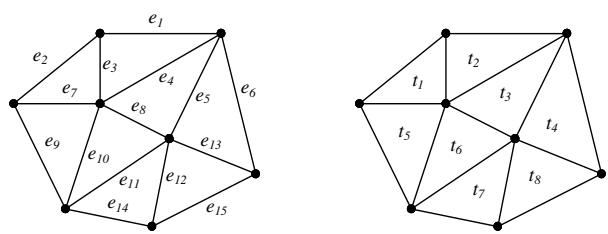


Fig. 1. Triangulation – a set of edges, a set of triangles

We mentioned that edges of the convex hull are always in the triangulation. Let us denote this kind of the edges as the common edges. However, the edges of convex hull are not alone in this group of edges. It can be expanded by extra edges according to the Definition 1. Then a general definition of the common edges can be following:

Definition 3. Let us assume that we have a set of different points in a plane $S = \{p_i\}$, $p_i \in E^2$, $i = 1, \ldots, N$ and complete undirected graph on this set of point $G = \{e_k : e_k = \{p_i, p_j\}, i \neq j, i, j = 1, \ldots, N\}$. The edges e_k from the graph G which cross no other edge are always in the arbitrary triangulation that can be constructed in the input set of points S and these edges are denoted as common edges.

Some examples of common edges for several sets of points are shown on the Fig. 2 (note that edges of convex hull also fulfill mentioned definition).

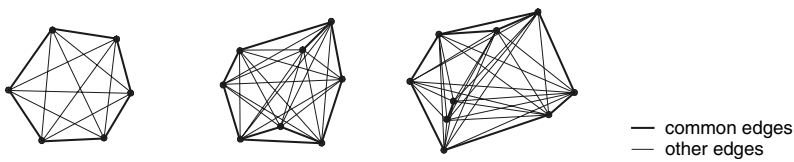


Fig. 2. Examples of common edges

The next important property is the theorem about a number of edges and triangles in an arbitrary triangulation that is possible to construct on a given set of points S.

Theorem 1. Let us assume that we have a set of N points $S = \{p_i\}$, $i = 1, \ldots, N$. If the number of points in the convex hull is N_{CH} then:

$$\begin{aligned} N_E &= 3 \cdot (N-1) - N_{CH} \\ N_T &= 2 \cdot (N-1) - N_{CH} \end{aligned}, \tag{1}$$

where N_E is the number of edges and N_T is the number of empty triangles in the triangulation.

Last property, which we can use, follows from the definition of the triangulation. No edges in the triangulation can cross mutually. Possibly, we can say about triangles that no two triangles in the triangulation can overlap more than in an edge.
This fact minimizes the number of edges that can be inserted into triangulation from the set of all possible edges. If an edge is inserted into a triangulation, we can be sure that all edges that cross this edge cannot be in the triangulation (this is also valid for triangles).

These three properties are valid for arbitrary sets of points. If we knew more about desired triangulations, we could find any extra properties (see [2], [5]). However, our goal is to design an algorithm which can be use for all kinds of the triangulations and which can find the result for all arbitrary criterions of the triangulation. Therefore, we will not think about this alternative.

3 Introduction about Triangulation Generating Methods

Generally, several approaches that solve the issue of searching for triangulations with a given properties exist. The ideal approach is based on usage algorithms with polynomial time complexity. However, those algorithms are only known for some properties of triangulation (e.g., Delaunay triangulation [1], [7]). In remaining cases a brute force algorithm has to be used. The brute force term means that all possible triangulations are generated, evaluated, and then the best one is selected. This approach is general and triangulations with arbitrary properties can be found. However, it also has a disadvantage. The algorithms generating all triangulations generally do not have polynomial time complexity (the NP problem [4], [6]) and, therefore, they only can find solutions on small sets of points. For all that, this paper is just dedicated to this approach and several algorithms are proposed in the following chapters. We will use knowledge from combinatorics [3], [8], [9] (combination generating and triangulation generating are similar problems) and knowledge about triangulations (see previous chapter) to design a fast, accurate and robust algorithm.
Note that one more approach exists. It is based on heuristic methods and can find some solutions for large sets of points. However, the triangulation found by this approach has not to be optimal. We only can be sure that it is an approximation of the exact solution with an error. This approach can be considered as a compromise between the polynomial time complexity and the exact solution.

3.1 Generator of Combination

From the equation (1) we know that all triangulations that can be constructed on a set of points still have the same number of edges NE. This fact and a generator of combinations together can be used to design an algorithm generating all triangulations as it is described in the following text.

If we made a unification of the edges from all the triangulations, which can be constructed, we would obtain a complete undirected graph of the set of points. Note that the maximal number of the edges in this graph is equal:

$$n = \binom{N}{2} = \frac{N \cdot (N-1)}{2}, \tag{2}$$

where N is the number of points.

Let us assign an index (from value *1* to n) to each edge in that complete undirected graph. Suppose also that a generator of combinations generates all possible sequences of N_E numbers where individual numbers are different mutually and they are from the range *1* to n. Then each combination can represent a triangulation and the number of those combinations is equal to the binomial coefficient of n and k that is defined as:

$$\binom{n}{k} = \frac{n!}{(n-k)!\cdot k!}, \; k = N_E - N_{CE}, \tag{3}$$

where n is the number of edges of the complete undirected graph (see the equation 2), N_E is the number of edges in triangulation (see the equation 1) and N_{CE} is the number of common edges.

This combinatorial number proofs that we can expect non polynomial time complexity. On the other hand, this is the worst case. Many combinations do not represent a triangulation because the condition of the crossing edges is not guaranteed. There is a question how to select the combinations representing the triangulations effectively. Two methods are possible:

A. All combinations are generated by very fast algorithm [3], [8], [9], and then the individual combinations are tested if they represent triangulations.
B. The algorithm is designed that it only generates the combinations of edges representing triangulations.

Theoretically, it is very hard to decide which of the methods is better. The first method uses a fast generator of the combinations. However, all combinations have to be generated and tested if they represent a triangulation. The second method only generates the combinations representing triangulations. However, the generator is slower because a test that excludes the unsuitable combinations is included in the generator. A threshold of the decision if it is better to use the A or B method affects many factors (the speed of generating combinations, the speed testing if a combination represents a triangulation, how many percents of combinations represent triangulations, etc.). Practically, it is more simple and infallible to implement the given algorithms and to compare them mutually as in our case. Note that a comparison of both methods is shown later in the chapter containing results.

3.2 Edge Removing Method

Complete undirected graph is remarked in the previous method. If we looked at the complete undirected graph again, we could find out that the unification of all edges, which are in the individual triangulations, also represents this graph. This fact is used in this method.

The start point of the algorithm is the complete undirected graph. When we will select and mark an edge in the graph as the edge that has to be in triangulation, we can remove all the edges that the given edge crosses. So we will obtain a new graph without any edges from the complete graph and with an edge that is marked as the edge of the triangulation. This procedure can be repeated until we obtain a graph that only includes edges representing a triangulation. Of course, we need to find all triangulations. The generating of the other triangulations is hidden in the mechanism of edge selecting that decides if individual edges have to be in the triangulation. This mechanism has to provide that no triangulation will be omitted and that any triangulations also will not be generated twice or more times.

The result structure that fulfils the requirements is a binary tree. The root of the tree represents the complete undirected graph and the leaves of the tree can be divided into two groups. In the first group, there are the leaves representing the triangulations according to the definition and, in the second one, there are the leaves that include non crossing edges, but their number is not adequate (see the equation 1).

Like in the previous algorithm, we have to assign the unique index to each edge. Then we can try to remove or to keep on the individual edges in the graph according to the index of the edges step by step. Each decision represents one level of the tree,

therefore, the maximal number of the levels is equal to the number of the edge in the complete graph (see the equation 2). However, this value is less in practice because the general properties of triangulations can be used in the implementation (see the chapter about the triangulation). An example of that tree with a binary vector representation is shown on the Fig. 3 (each bit represents one edge with a given index, the value '1' means that the edge is in the graph).

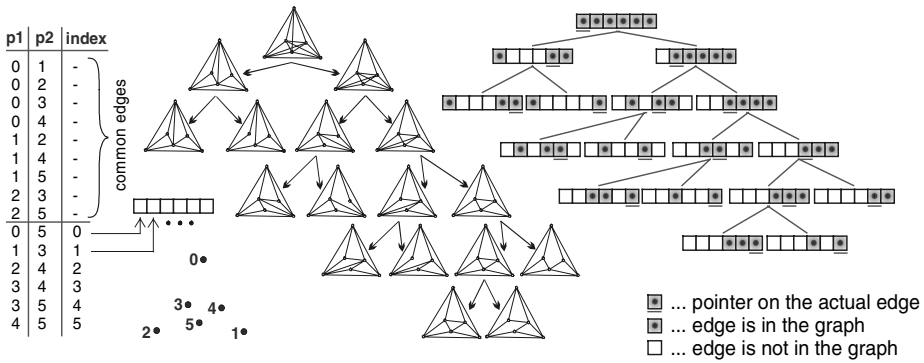


Fig. 3. An example of the edge removing method

3.3 Edge Inserting Method

This method is very similar to previous one. The only main difference is that the root node of the tree represents the complete graph but the so called empty graph (it means the graph with no edges). Otherwise, the algorithm is the same. A good question is if this method towards the previous method is faster. Theoretically, it is very hard to decide. It is affected many factors and, therefore, the implementation on the given kind of the problem is the infallible way. An example of the tree with the representation by a binary vector (like in the previous method) is shown on Fig. 4.

It perhaps seems that a representation by the binary vector is not possible. It is not true. When we select an arbitrary node in the graph, we can separate the binary vector into two parts (the left and right part) by the pointer on an actual edge. The bits of the left part represent the edges which have been in process and their status only indicates that the edges are or are not in the triangulation. The bits in the right part of the binary vector (inclusive of the actual edge) represent the edges which have not been in process yet and their status can say if the given edge still can be inserted into the triangulation or if it is not possible. Now, it is sure that the binary representation is sufficient and suitable in this case.

3.4 Triangle Inserting Method

In this last method that is introduced here we look on a triangulation like on a set of the triangles. Of course, we could look at the triangulation from the same view in previous methods and we could work with the empty triangles instead of the edges. However, this approach would be worse and the final algorithm would be slower.

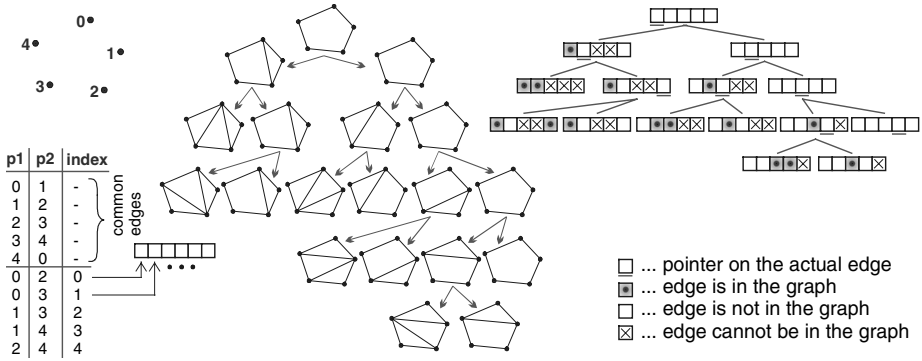


Fig. 4. An example of the edge inserting method

Let us return to our algorithm. At the beginning of this paper we said that the convex hull was in all the triangulations. We used this fact here and the convex hull is the start point of this algorithm. Exactly, the convex hull represents a polygon (the so called *boundary polygon*) surrounding a region into which triangles have to be inserted for creating a correct triangulation. The procedure of the algorithm is very simple. An edge is chosen from the boundary polygon, and then the so called empty triangle is inserted if it contains the selected edge and if it is inside the boundary polygon. The empty triangle means a triangle whose vertices are any points from the input set and which contains no other points from this set (see definition 2). By inserting the triangle, the boundary polygon will be changed and will demark the original region without the region of the inserted triangle. From this new polygon an edge is selected and another empty triangle, which contains this selected edge and which is included inside the new region, is inserted again. That procedure is repeated until a correct triangulation is created (the boundary polygon just represents an empty triangle).

Now we obtain one triangulation, however, we need to generate all triangulations. It is possible to generate them when we ensure inserting all combinations of the empty triangles for the given selected edge. We will get a tree data structure where the root is the node including the edge of the convex hull and where the leaves of the tree represent the triangulations. Each intermediate node has as many branches as many empty triangles can be inserted for the selected edge of the given boundary polygon. An example of this tree is shown on the Fig. 5.

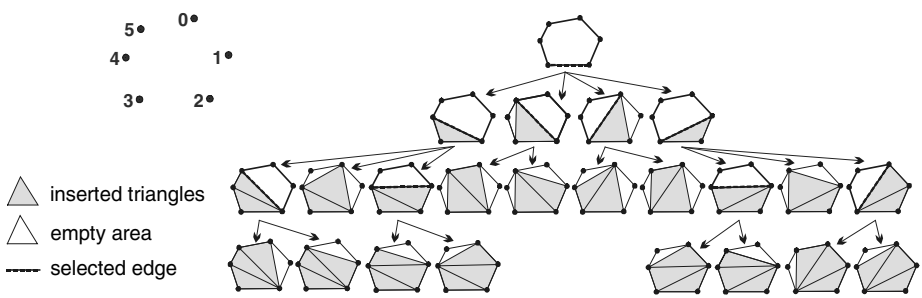


Fig. 5. An example of the triangle inserting method

4 Results

We described a few algorithms that generate all triangulations. In this chapter we compared them mutually. The described algorithms were implemented for the MWT (*Minimum Weight Triangulation*) [5], [7], [10] where weights of edges are represented by Euclidian distances between the end points of edges. For this criterion, an algorithm with a polynomial time complexity still has not been found, therefore, it is an ideal situation for testing mentioned algorithms.

Our goal is to find the triangulation that has a minimal sum of weights of edges. A structure of programs with the individual algorithms is similar and simple. When a triangulation is found, it is evaluated and tested (the main task of the test is to remember the triangulation with the best evaluation). When all possible triangulations are found, we can be sure that we have obtained the best one. Note that an advantage of this approach is in a small memory requirement and we always find the global optimal solution. We do not need to remember all triangulations but only the best one. We tested all algorithms for randomized generated sets of points on the same computer (DELL, 450 MHz, 1 GB RAM) with OS Windows 2000. The resultant graph that characterizes the time dependence of the calculation on the number of points is shown on the Fig. 6.

The values in the graph were calculated as an average of times that had been measured for the sets with the same number of points. Consequently, the values in the graph are only expected times that were measured for the given kind of data (the uniform distribution of points in a plane) on the given computer. For all that, we can obtain some basic information about the individual algorithms and we can determine which method is faster or slower. We can obtain an estimation of time for evaluation of a bigger set of points, etc.

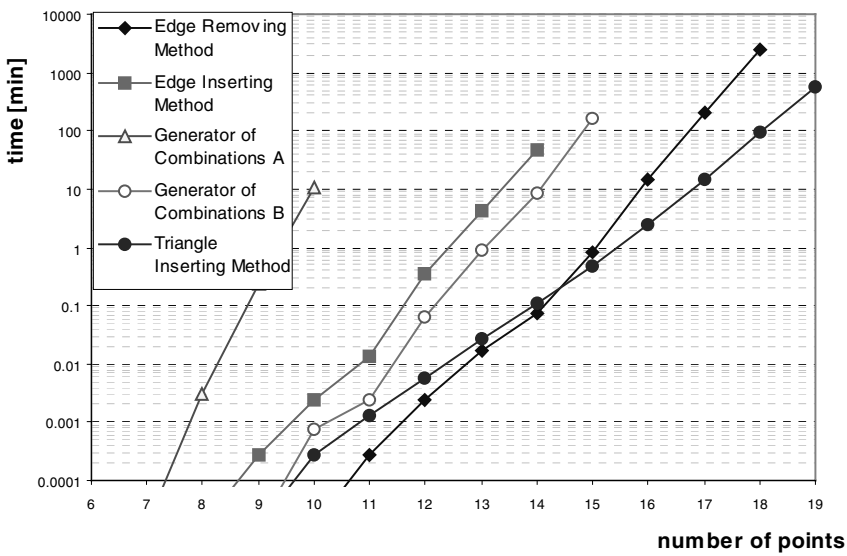


Fig. 6. The graph that shows the expected time needed to finding for the MWT by the designed methods (a dependence on the number of points)

We can also estimate time complexity of the algorithms for another criterion on the triangulation. The test searching for the MWT has *O(N)* time complexity in the algorithm (the sum of edge weights has to be calculated for the found triangulation). When we select the criterion that has the same time complexity for criterion evaluation in the algorithm, we can use these results to estimate of needed time for calculation.

5 Conclusion

The main goal of this work was to generate optimal triangulations for a required criterion. It is expected that such generated triangulations will be used for verification of new algorithms and for effective triangulation generating.

This paper presents an overview of new approaches. Several methods searching for global optimal triangulations with required properties were developed, implemented and tested. The comparison of developed algorithms generating all possible triangulations was also made. By comparing the individual curves in the graph (see the Fig. 6), we can see properties of developed algorithms. Generally, the complexity of the triangular mesh generator is not polynomial and, therefore, a selection of an unsuitable data structure or algorithm influences extensively the time that is needed for the computation.

Finally, note that although the algorithms are designed for a triangulation generator, the introduced algorithms can also be used to solve similar problems (e.g., combination generating, etc.).

References

1. Aurenhammer, F.: Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys 23(3): 345–405, 1991.
2. Drysdale, R., L., S., McElfresh, S., Snoeyink, J., S.: An improved diamond property for minimum weight triangulation. 1998.
3. Ehrlich, G.: Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. Journal of the ACM, vol. 20, Issue 3, pp. 500–513, 1973.
4. Garey, M., R., Johnson, D., S.: Computers and Intractability: A Guide to the theory of NP-completeness. W. H. Freeman, San Francisco, 1979.
5. Jansson, J.: Planar Minimum Weight Triangulations, Master's Thesis, Department of Computer Science, Lund University, Sweden, 1995.
6. Kucera, L.: Combinatorial Algorithms, ISBN 0-85274-298-3, SNTL, Publisher of Technical Literature, 1989.
7. Preparate, F. P., Shamos, M. I.: Computational Geometry - an Introduction, Springer-Verlag, New York, 1985.
8. Takaoka, T.: O(1) time algorithms for combinatorial generation by tree traversal. Computer Jurnal, vol. 42, no. 5, pp. 400–408, 1999.
9. Xiang, L., Ushijima, K.: On O(1) Time Algorithms for Combinatorial Generation. The Computer Journal, vol. 44, no. 4, pp. 292–302, 2001.
10. Yang, B., T., Xu, Y., F., You, Z., Y.: A chain decomposition algorithm for the proof of a property on minimum weight triangulations. 1994.

Approximations for Two Decomposition-Based Geometric Optimization Problems*

Minghui Jiang, Brendan Mumey, Zhongping Qin,
Andrew Tomascak, and Binhai Zhu

Department of Computer Science, Montana State University,
Bozeman, MT 59717-3880, USA.
{jiang,mumey,qin,tomascak,bhz}@cs.montana.edu

Abstract. In this paper we present new approximation algorithms for two NP-hard geometric optimization problems: (1) decomposing a triangulation into minimum number of triangle strips (tristrips); (2) covering an $n \times n$ binary neuron image with minimum number of disjoint $h \times h$ boxes such that the total number of connected components within individual boxes is minimized. Both problems share the pattern that overlap is either disallowed or to be minimized. For the problem of decomposing a triangulation into minimum number of tristrips, we obtain a simple approximation with a factor of $O(\sqrt{n \log n})$; no approximation with $o(n)$ factor is previously known for this problem [6]. For the problem of tiling a binary neuron image with boxes, we present a bi-criteria factor-$(2, 4h-4)$ approximation that uses at most twice the optimal number of tiles and results in at most $4h-4$ times the optimal number of connected components. We also prove that it is NP-complete to approximate the general problem within some fixed constant.

1 Introduction

In this paper, we present efficient approximation algorithms for two geometric optimization problems arising in computer graphics and computational biology. Both problems are NP-hard. Although our techniques in designing approximation algorithms for these problems are somehow standard, we believe that some properties of these problems we prove in this paper might have applications for other problems.

Decomposing a triangulation into triangle strips (tristrips) has interesting application in computer graphics and visualization. It is an open problem posed by Rossignac. Recently Estkowski et al. [6] proved that this problem is NP-complete; they also proposed two algorithms, with no proven approximation factor, for this problem.

In this paper, we first present a simple linear time algorithm to decide whether a triangulation can be encoded by a single tristrip; we then present a factor-$O(\sqrt{n \log n})$ approximation algorithm for the general optimization problem. Our

* This research is partially supported by NSF CARGO grant DMS-0138065.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 90–98, 2004.
© Springer-Verlag Berlin Heidelberg 2004

approximation is achieved by first approximating the related problem of *covering* a triangulation with minimum number of tristrips using set cover.

Tiling (decomposing) an image with fixed-size boxes has a lot of applications in data partition, image storage, and statistical analysis. Given a set of points in 2D, covering them using the minimum number of fixed-size boxes (squares) was proved be NP-complete more than two decades ago [7]. Recently, Khanna, *et al.* studied a slightly different problem, namely, to partition an $n \times n$ array of non-negative numbers into minimum number of tiles (fixed-size rectangles) such that the maximum weight of any tile is minimized. (Here the *weight* of a tile is the sum of the elements within it.) They proved that this problem is NP-complete and that no approximation can achieve a factor better than 1.25; they also proposed a factor-2.5 approximation algorithm [11]. Improved bounds are obtained later [14,3]. Some related problems are studied in [2].

In this paper, we study another related problem originated from the application of storing and manipulating neuron images in computational biology. In the study of neural maps, biologists need to divide a large 3D neuron image (represented by stacks of 2D images) into disjoint fixed-size boxes and store them separately [9,10,13]. The size of each box is much smaller and its data can be handled by a common computer, say, a PC. In the decomposition of a neuron image, we want to keep enough information within each box, which stores fragments of a neuron, while using a limited number of boxes: we want to minimize the total number of connected components with individual boxes and, at the same time, bound the number of boxes used. We formulate this problem as a more general 2D problem. Given an $n \times n$ binary image M, decide whether the 1-elements in M can be covered by B disjoint $h \times h$ ($h \geq 2$) boxes such that the total number of connected components within individual boxes is bounded by W. We show that this problem is NP-complete. We also present a bi-criteria factor-$(2, 4h-3)$ approximation for this problem. When all the 1-elements in the image are connected, as is the actual situation in the applications since a neuron is basically a huge tree in 3D, then the approximation factor becomes $(2, 4h-4)$. Our approximation results can be generalized straightforwardly to 3D but with higher approximation ratios.

We now define formally the problems to be studied. We first make some definitions related to our algorithms. As these problems are NP-hard, from now on we will focus on their *optimization versions*. We say that an approximation algorithm for a maximization (minimization) problem Π provides a *performance guarantee* of ρ if, for every instance I of Π, the solution value returned by the approximation algorithm is at least $1/\rho$ (at most ρ of) the optimal value for I. For simplicity, we also say that this is a factor-ρ approximation algorithm for Π.

We first introduce the concept of a *triangle strip* (tristrip). A triangulation T with n triangles is a tristrip if there exists a vertex sequence $\langle v_1, v_2, ..., v_{n+2} \rangle$, possibly with repetition, such that the n triangles of T are exactly given by the triples of the consecutive vertices of the sequence. A tristrip s_1 *overlaps* another tristrip s_2 if a triangle is encoded in both s_1 and s_2; otherwise, we say that s_1 and s_2 are *disjoint*. In Fig. 1, we show two overlapping tristrips: $s_1 = \langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \rangle$ and $s_2 = \langle 12, 11, 6, 4, 5, 3, 14, 13 \rangle$. Finding the minimum

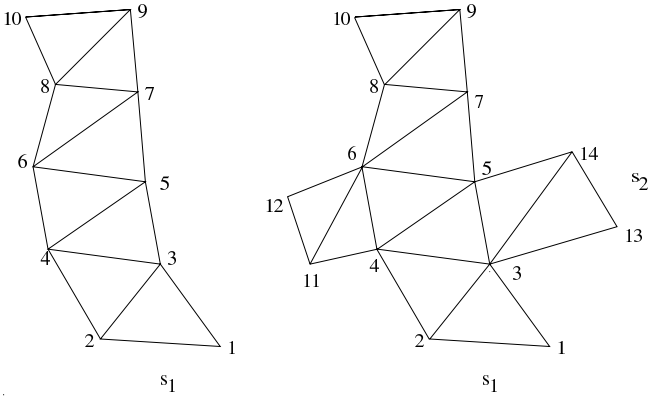


Fig. 1. An example of two tristrips.

number of (disjoint) tristrips that encode T is an interesting research topic, since this can reduce the transmit and rendering time in graphics and visualization. The k-STRIPABILITY problem is defined as follows:
Instance: Given a triangulation T of n triangles, a positive integer k.
Problem: Does there exist k disjoint tristrips whose union is exactly T?

As this problem is NP-complete [6], we will try to find an approximation algorithm for it (approximating the smallest k).

Our second problem *Geometric Tiling with Fixed-size Boxes to Minimize Connected Components* (BOX-TILE-#CC) is defined as follows:
Instance: Given a $n \times n$ binary array M, integers $B, W > 0$ and $2 \leq h \leq n$.
Problem: Does there exist at most B disjoint $h \times h$ boxes that cover all the 1's in M and the total number of connected components within individual boxes is at most W?

Note that in this problem we have two criteria for optimization: the number of boxes B and the total number of connected components W (W stands for "Weight"). Let the optimal solutions of BOX-TILE-#CC for these two criteria be OPT_B and OPT_W respectively. We say that a bi-criteria approximation algorithm provides a *performance guarantee* of (α, β) for BOX-TILE-#CC if for every instance I of BOX-TILE-#CC, the solution value returned by the approximation algorithm uses at most $\alpha \times OPT_B$ boxes and the total number of connected components within these boxes is at most $\beta \times OPT_W$.

In the next two sections, we present details of our approximations for these two problems.

2 Approximation Algorithm for Decomposing a Triangulation into Tristrips

In this section we will present an approximation algorithm for computing the minimum number of tristrips that encode T. Recall that the k-STRIPABILITY

problem is defined as deciding whether T can be partitioned to a minimum number of k tristrips. Our approximation is based on an approximate solution for a slightly different problem, namely, finding the minimum number of tristrips that *cover* a triangulation T. We call the latter problem *Minimum Strip Covering* (MSC). Clearly, in MSC two tristrips might overlap.

We first discuss the special case for k-STRIPABILITY when $k = 1$. It is easy to have the following lemma.

Lemma 1. *Given any triangulation T, we can decide the 1-STRIPABILITY of T in linear time.*

Proof. Pick an arbitrary triangle $\triangle v_i v_j v_k$ from T. From the definition of tristrip, we can clearly see that, if there exists a single tristrip that encodes T, then at least one of three subsequences $v_i v_j v_k$, $v_j v_k v_i$, and $v_k v_i v_j$, or its reversal, has to appear in the tristrip sequence. For each subsequence, we can "grow" it from both ends to recover the whole sequence. It is crucial to notice that, at each step of the growth, the next vertex to visit is automatically decided by the previous two vertices; therefore it takes linear time recover the tristrip sequence if it exists. □

We now present the details of a factor-$O(\log n)$ approximation algorithm for the MSC problem. Our approximation uses the same greedy method for approximating the Set Cover problem, so we first briefly introduce the Set Cover problem, which is defined as follows: Given a set X, and a set F which is a family of subsets of X, find a subset C of F with minimum cardinality such that every member of X is contained in at least one member of C (which is a subset of X). It was proved [4,8,12] that, by using a greedy method to find each maximal subset that covers the maximum number of uncovered elements, an $O(\log |X|)$-factor approximation algorithm can be obtained.

We can look at the MSC problem from another perspective: a triangulation T is a set X of triangles, and a tristrip is a subset of X. Our goal is to find the smallest subset C from the family of all tristrips F, such that every triangle in X is contained in at least one tristrip in C. This is exactly a Set Cover problem, for which a factor-$O(\log |X|)$ approximation algorithm is already known.

In order to implement the set cover algorithm, we need to decide a proper X and F. X is naturally the set of all the triangles in the triangulation. F has to contain all the possible tristrips in the triangulation and seems difficult to compute; however, since two tristrips are allowed to overlap in MSC, we only need to consider maximal tristrips (i.e., no tristrip is completely contained in another). Starting from each triangle in T, we can find at most three maximal tristrip, using the method outlined in Lemma 1; we collect these tristrips in set F. We then apply the greedy method to find an approximate minimum cover A.

Let the optimal solution value for the MSC problem be C^* and let the optimal solution value for the k-STRIPABILITY problem be K^*. Our approximation for MSC has a solution value of $|A|$, which satisfies $|A| \leq O(\log n) \cdot C^*$; in other words, we can cover T using at most $O(\log n) \cdot C^*$ tristrips.

To convert our approximation for MSC to an approximation for k-STRIPABILITY, first note that $C^* \leq K^*$. The reason is that a decomposition solution is always a solution for the corresponding covering problem but not vice versa. Second, it is easy to see that a solution of $|A|$ tristrips for MSC can be converted to $O(|A|K^*)$ disjoint tristrips. For any two tristrips t_1 and t_2 from A such that t_1 overlaps t_2, we decompose them into disjoint sets of tristrips: $t_1 - t_1 \cap t_2$, $t_2 - t_1 \cap t_2$, and $t_1 \cap t_2$ ($t_1 \cap t_2$ might contain more than one intersection). For each intersection in $t_1 \cap t_2$, we must use at least one tristrip in the optimal solution for k-STRIPABILITY, i.e., each tristrip in our approximation solution for MSC can be decomposed into at most $O(K^*)$ pieces. Therefore, this decomposition from the $|A|$ tristrips, possibly overlapping, introduces at most $O(|A|K^*)$ disjoint tristrips, which presents a natural solution for k-STRIPABILITY.

To determine the approximation factor, we check the approximation A obtained for MSC. If $|A| \geq \log n \cdot n^x$ (x is to be determined), then $\log n \cdot n^x \leq |A| \leq O(\log n)C^* \leq O(\log n)K^*$, hence $K^* \geq C^* \geq c_1 n^x$ (c_1 is some constant). As the converted approximation for k-STRIPABILITY is at most $O(|A|K^*) = O(n)$, the approximation factor is at most $\frac{O(n)}{c_1 n^x} = O(n^{1-x})$ in this case. Otherwise, if $|A| \leq \log n \cdot n^x$, then $|A|K^* \leq \log n \cdot n^x \cdot K^*$. Therefore, the approximation factor for the latter case is at most $O(\log n \cdot n^x)$. To obtain the right x, we set $n^{1-x} = \log n \cdot n^x$, which gives us $x = \frac{1}{2} - \frac{\log \log n}{2 \log n}$. Consequently the overall approximation factor of this algorithm is $O(\sqrt{n \log n})$.

It is clear that the running time of our approximation algorithm is $O(n^2)$. We summarize our result in the following theorem.

Theorem 1. *Given any triangulation T with n triangles, there is a factor-$O(\sqrt{n \log n})$ approximation that runs in $O(n^2)$ time for the k-STRIPABILITY problem.*

It is interesting to see whether this approximation factor can be further improved. The graph version of this problem (interesting enough, the next problem too!) is not approximable within a factor of $O(|V|^\theta)$, $0 < \theta < 1$, unless P=NP: Given a set of $|V|$ red and blue intervals, and the corresponding intersection graph G with $|V|$ vertices, the problem of computing the smallest number of independent red vertices that dominate all blue vertices cannot be approximated within a factor of $O(|V|^\theta)$, $0 < \theta < 1$, unless P=NP [5]. Of course, our problem contains extra geometric information which probably explains why this $O(\sqrt{n \log n})$ approximation factor is achievable.

3 Fixed-Size Geometric Tiling to Minimize Connected Components

Given a binary matrix, we first consider a slightly different problem called BOX-TILE, namely, the problem of covering the 1-elements in M using the minimum number of disjoint fixed-size boxes. It is shown [7] that BOX-TILE is NP-complete. Let the optimal solution of BOX-TILE be $OPT_{\#}$. We will first present an approximation for BOX-TILE with the following lemma.

Lemma 2. *There is an $O(M)$ time, factor-2 approximation for BOX-TILE.*

Proof. We use a *striping* method. First consider the first h rows of M and use a simple linear scan to cover all the one's in them such that no two boxes overlap. This is essentially a 1D problem and can be solved optimally by greedy method. We then repeat this process every h rows until all the one's in M are covered. Let the minimum number of boxes used for each strip be N_i, $i = 1, 2, \ldots, \lceil n/h \rceil$; the total number of boxes used is $\sum_i N_i$.

Clearly, every box in an arbitrary tiling for BOX-TILE intersects at most two strips. If we duplicate each box in the optimal solution that intersects two strips, and push one copy into the upper strip and the other copy into the lower strip, the result is a valid covering. We can then rearrange the boxes in each strip to avoid overlapping and obtain a tiling, the total number of boxes in which is at least $\sum_i N_i$. Therefore, we have

$$\sum_i N_i \leq 2OPT_{\#}.$$

□

Notice that if we are willing to increase the running time of our approximation, BOX-TILE can be approximated within factor $(2 - \frac{1}{h})$ of optimal. Without loss of generality, we assume that the 1 entries are at least h cells away from the upper and lower boundaries of M. The basic idea is to use h *shifted* applications of the striping method. For each offset from 1 to h, we generate a shifted striping pattern: the pattern for offset 1 has strip rows $[1, h]$, $[h+1, 2h]$, ...; the pattern for offset 2 has strip rows $[2, h+1]$, $[h+2, 2h+1]$, ...; and so on. Let $APP_{\#}$ be the minimum total number of tiles among our approximation solutions for these patterns. We claim that

$$APP_{\#} \leq \lfloor (2 - \frac{1}{h}) OPT_{\#} \rfloor.$$

To prove our claim, consider the $OPT_{\#}$ tiles in the optimum solution of BOX-TILE. By the Pigeonhole principle, at least one shifted striping pattern is guaranteed to have at least $\lceil OPT_{\#}/h \rceil$ tiles that fall exactly with its strips. Applying the doubling procedure to the remaining tiles yields a solution to the strip-restricted problem using at most $\lfloor (2 - \frac{1}{h}) OPT_{\#} \rfloor$ tiles. Since $APP_{\#}$ is optimal over all strip-restricted sub-problems, this finishes the proof for the following lemma.

Lemma 3. *There is an $O(hM)$ time, factor-$(2 - \frac{1}{h})$ approximation for BOX-TILE.*

We now describe our approximation for BOX-TILE-#CC. For simplicity, in the following we always use the simple one-round striping method (Lemma 2) as a subroutine; clearly, all the following results can be improved if the h-round striping method (Lemma 3) is used instead.

Theorem 2. *There is a $(2, 4h-3)$-approximation for BOX-TILE-#CC.*

Proof. We simply use the result in Lemma 2 as our approximation for OPT_B. As $OPT_{\#} \leq OPT_B$, we have

$$\sum_i N_i \leq 2OPT_{\#} \leq 2OPT_B.$$

We now consider the connected components within each box in our approximation. Since a $h \times h$ box has $4h-4$ boundary elements, we have at most $2h-2$ connected components that "touch" the boundary, i.e., every other element along the boundary is a 1-element and each belongs to a different connected component within the box. The total number of these connected components is at most

$$(2h-2)\sum_i N_i \leq (4h-4)OPT_{\#} \leq (4h-4)OPT_B.$$

As $OPT_W \geq OPT_B$, we have

$$(2h-2)\sum_i N_i \leq (4h-4)OPT_W.$$

For those connected components not touching the box boundaries, it is easy to see that their total number is at most OPT_W, since each of them contributes at least one in the total number of connected components in any tiling. Therefore, the total number of connected components in our approximation is at most $(4h-3)OPT_W$. □

In the case when the 1's in M are connected, every connected component in any tiling must touch the box boundaries (except for the uninteresting case where all the 1's in M can be covered by a single box, which can be found by our h-round method outlined in Lemma 3). Therefore we have the following corollary.

Corollary 1. *If the 1-elements in M are connected, then there is a $(2, 4h-4)$ approximation for BOX-TILE-#CC.*

We believe that for most data sets encountered in practice our algorithm actually presents a better approximation, as implied in corollary above. This has been partially verified by some empirical results [15] obtained on small-size random data (whose optimal solutions can be computed). We plan to obtain empirical results over real data actually used in practice and the details will be presented later. Finally, we summarize the hardness result for BOX-TILE-#CC. Due to the space limitation, the proof will be covered in the final version of this paper.

Theorem 3. *BOX-TILE-#CC is NP-complete; moreover, it is impossible to obtain a γ-approximation for OPT_W, where $\gamma > 1$ is some fixed constant, unless P=NP.*

Note that if the boxes are allowed to have overlaps then we are again back to the weighted SET-COVER problem, and a factor-$O(\log h)$ approximation is easy

to obtain [4]. We believe that, with a simple heuristic, we might obtain better approximations for both BOX-TILE and BOX-TILE-#CC. The idea is to check pairs of neighboring boxes from adjacent strips (as computed from Lemma 2 or 3), and determine whether the 1-elements in both boxes can be jointly covered by a single box (across two strips).

4 Concluding Remarks

In this paper we obtain a factor-$O(\sqrt{n \log n})$ approximation for the k-STRIPABILITY problem and a bi-criteria factor-$(2, 4h - 4)$ approximation for the BOX-TILE-#CC problem. Several questions remain unanswered: (1) For the k-STRIPABILITY problem, whether we can have an approximation with a factor significantly better than $O(n^{1/2})$ is still open. In fact, little is known about the topological characteristic of an optimal solution for k-STRIPABILITY. So far we only know that a tristrip is formed by making alternative left and right turns along edges of the input the triangulation. More study is necessary for this problem. Another related open problem for k-STRIPABILITY is to decompose a triangulation T with the minimum number of Hamiltonian triangulations [1]. (2) For the BOX-TILE-#CC problem, when the image stored in M is a tree, can we exploit this fact to either improve the approximation factor further or show that such an improvement is impossible?

As we have mentioned at the end of Section 2, for the graph versions of the k-STRIPABILITY and BOX-TILE-#CC problems, the corresponding problem of Minimum Independent Dominating Set on Bichromatic Circle Graphs is not approximable within a factor of $O(|V|^{\theta})$, $0 < \theta < 1$, unless P=NP [5]. So our approximation factors for k-STRIPABILITY and BOX-TILE-#CC (when $h = o(n)$) have already broken this barrier. Of course, our problems contain extra geometric information and are not exactly the same as the corresponding graph problems.

Acknowledgment. We thank Xun He for discussion on the k-STRIPABILITY problem.

References

1. Esther M. Arkin, Martin Held, Joseph S. B. Mitchell, and Steven Skiena. Hamiltonian triangulations for fast rendering. *The Visual Computer*, 12(9):429–444, 1996.
2. Piotr Berman, Bhaskar DasGupta, and S. Muthukrishnan. Slice and dice: a simple, improved approximate tiling recipe. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms (SODA'02)*, pages 455–464, 2002.
3. Piotr Berman, Bhaskar DasGupta, S. Muthukrishnan, and Suneeta Ramaswami. Improved approximation algorithms for rectangle tiling and packing. In *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pages 427–436, 2001.
4. Vašek Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4:233–235, 1979.

5. Mirela Damian-Iordache and Sriram V. Pemmaraju. Hardness of approximating independent domination in circle graphs. In *Proc. 10th Annual International Symposium on Algorithms and Computation (ISAAC'99)*, LNCS 1741, pages 56–69, 1999.
6. Regina Estkowski, Joseph S. B. Mitchell, and Xinyu Xiang. Optimal decomposition of polygon models into triangle strips. In *Proc. 18th ACM Symposium on Computational Geometry (SoCG'02)*, pages 254–263, 2002.
7. Robert J. Fowler, Mike Paterson, and Steven L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information Processing Letters*, 12(3):133–137, 1981.
8. David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences* 9(3):256–278, 1974.
9. G. Jacobs and F. Theunissen. Functional organization of a neural map in the cricket cercal sensory system. *Journal of Neuroscience*, 16(2):769–784, 1996.
10. G. Jacobs and F. Theunissen. Extraction of sensory parameters from a neural map by primary sensory interneurons. *Journal of Neuroscience*, 20(8):2934–2943, 2000.
11. Sanjeev Khanna, S. Muthukrishnan, and Mike Paterson. On approximating rectangle tiling and packing. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 384–393, 1998.
12. László. Lovász. On the ratio of optimal integral and fractional covers, *Discrete Mathematics* 13:383–390, 1975.
13. S. Paydar, C. Doan, and G. Jacobs. Neural mapping of direction and frequency in the cricket cercal sensory system. *Journal of Neuroscience*, 19(5):1771–1781, 1999.
14. Adam Smith and Subhash Suri. Rectangular tiling in multi-dimensional arrays. In *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 786–794, 1999.
15. Andrew Tomascak. *Fixed-size Geometric Covering to Minimize the Number of Connected Components*. M.Sc. Thesis, Department of Computer Science, Montana State University, 2003.

Computing Largest Empty Slabs

Jose Miguel Díaz-Báñez[1], Mario Alberto López[2], and Joan Antoni Sellarès[3]

[1] Universidad de Sevilla, SPAIN,
dbanez@us.es
[2] University of Denver, USA,
mlopez@cs.du.edu
[3] Universitat de Girona, SPAIN
sellares@ima.udg.es

Abstract. Let S be a set of n points in three-dimensional Euclidean space. We consider the problem of positioning a plane π intersecting the convex hull of S such that $\min\{d(\pi, p); p \in S\}$ is maximized. In a geometric setting, the problem asks for the widest empty slab through n points in space, where a slab is the open region of $\mathbb{R}^3$ that is bounded by two parallel planes that intersect the convex hull of S. We give a characterization of the planes which are locally optimal and we show that the problem can be solved in $O(n^3)$ time and $O(n^2)$ space. We also consider several variants of the problem which include constraining the obnoxious plane to contain a given line or point and computing the widest empty slab for polyhedral obstacles. Finally, we show how to adapt our method for computing a largest empty annulus in the plane, improving the known time bound $O(n^3 \log n)$ [6].

1 Introduction

Location science is a classical field of operations research that has also been considered in the computational geometry community. A class of problems from this field, often referred to as maximin facility location, deals with the placement of undesirable or obnoxious facilities. In these problems the objective is to maximize the minimal distance between the facility and a set of input points. Furthermore, in order to ensure that the problems are well-defined the facility is normally constrained to go through some sort of bounding region, such as the convex hull or bounding box of the input points. Applications of these problems go well beyond the field of location science. For instance, splitting the space using cuts that avoid the input points is useful in areas like cluster analysis, robot motion-planning and computer graphics.

Maximin facility location problems have recently been considered in computational geometry. Maximin criteria have been investigated in 2-d for the optimal positioning of lines [10], anchored lines [9], and circumferences [6]. When the facility is a line, the problem is equivalent to that of computing a *widest empty corridor*, i.e., a largest empty open space bounded by two parallel lines. Variants of the problem have also been considered and include corridors containing k input points [11,13,4], dynamic updates [11,13] and L-shaped corridors [5]. Most

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 99–108, 2004.
© Springer-Verlag Berlin Heidelberg 2004

of the results to date are two-dimensional and, with a few exceptions (e.g., [9]), little progress has been reported in three dimensions.

In this paper, we deal with the maximin location of a plane in 3-d. We formulate the *obnoxious plane problem*, **OPP**, as follows: Given a set S of n points in $\mathbb{R}^3$, find a plane π intersecting the convex hull of S which maximizes the minimum Euclidean distance to the points.

Notice that, in 2-d, our problem reduces to that of computing the *widest empty corridor* through a set of points in the plane. This problem has been solved in $O(n^2)$ time and $O(n)$ space [10]. We extend the definition of corridor through a point set from $\mathbb{R}^2$ to $\mathbb{R}^3$ as follows: a *slab* through S is the open region of $\mathbb{R}^3$ that is bounded by two parallel planes that intersect the convex hull of S. The *width* of the slab is the distance between the bounding planes. Thus, we are interested in finding *the widest empty slab.*

The rest of the paper is organized as follows. In Section 2, we present some notation and preliminary results. In Section 3, we describe an algorithm to compute an obnoxious plane in $O(n^3)$ time and $O(n^2)$ space. Other variants, obtained by constraining the optimal plane π to go through a given line or given point, are described in Section 4, and solved in $O(n \log n)$ and $O(n^{2+\varepsilon})$ time, respectively. In Section 5, we compute the widest empty slab through a set of polyhedral obstacles within the same bounds as the **OPP**. Finally, section 6 presents a reduction of the largest empty annulus problem to our problem.

2 Characterization of Candidate Planes

In this section we describe a simple formula to compute the width of a slab and derive necessary conditions for slab optimality.

Observation 1. Let π and σ be two distinct parallel planes with (common) unit normal $\overrightarrow{n}$. Let p and q be arbitrary points on π and σ, respectively. Then, $\text{dist}(\pi, \sigma) = |\overrightarrow{n} \cdot (q - p)|$.

Lemma 1. *Let π^* be a solution to an instance of* **OPP** *and let π_1 and π_2 be the bounding planes of the slab generated by π^*. Then, exactly one of the following conditions must hold:*

(a) Each of π_1 and π_2 contains exactly one point of S, p_1 and p_2 respectively, such that $p_2 - p_1$ is orthogonal to π^*.
(b) There are points $S_1 = \{p_{11}, \ldots, p_{1h}\} \subset S$ on π_1 and $S_2 = \{p_{21}, \ldots, p_{2k}\} \subset S$ on π_2 such that $h \geq 2$, $k \geq 1$ and $S_1 \cup S_2$ lie on a common plane τ that is orthogonal to π^*.
(c) There are points $S_1 = \{p_{11}, \ldots, p_{1h}\} \subset S$ on π_1 and $S_2 = \{p_{21}, \ldots, p_{2k}\} \subset S$ on π_2 such that $h \geq 3$, $k \geq 1$, S_1 are not collinear, and $S_1 \cup S_2$ are not coplanar.
(d) There are points $S_1 = \{p_{11}, \ldots, p_{1h}\} \subset S$ on π_1 and $S_2 = \{p_{21}, \ldots, p_{2k}\} \subset S$ on π_2 such that $h \geq 2$, $k \geq 2$, S_1 are collinear, S_2 are collinear, and $S_1 \cup S_2$ are not coplanar.

Proof. We begin with the obvious observation that both π_1 and π_2 must contain at least one point of S as, otherwise, $\mathrm{dist}(\pi_1, \pi_2)$ can be increased. In the sequel, let $\overrightarrow{n}$ be a unit normal to π^* (hence, also normal to π_1 and π_2) chosen so that $(q-p)\cdot\overrightarrow{n} > 0$ for any points q on π_2 and p on π_1. Conceptually, we find π_2 (resp. π_1), by translating a copy of π^* in direction $\overrightarrow{n}$ (resp. $-\overrightarrow{n}$), parallel to itself, until at least one point of S is encountered. The cases described in the lemma exhaustively cover all possibilities for the number of points encountered when performing this translation.

First, consider case (a). Suppose π^* is not orthogonal to $p_2 - p_1$. Then π_1 and π_2 can be rotated simultaneously around p_1 and p_2, respectively, so as to decrease the angle between $\overrightarrow{n}$ and $p_2 - p_1$, while keeping the slab empty. This, in turn, increases $\overrightarrow{n}\cdot(p_2 - p_1) = \mathrm{dist}(\pi_1, \pi_2)$, contradicting the optimality of π^*.

Consider now case (b) and assume that the plane τ through p_{11}, p_{12} and p_{21} is not orthogonal to π^*, so that the angle ϕ between $\overrightarrow{n}$ and τ is strictly positive. We show that a small rotation of π_1 around the line $\overline{p_{11}p_{12}}$ (and a simultaneous rotation of π_2 around p_{12} that keeps the two planes parallel) can be performed so as to decrease the angle between $\overrightarrow{n}$ and $p_{21} - p_{11}$ while keeping the slab empty. In order to assess the effect of the rotation let $\overrightarrow{u}$ denote a unit normal to the rotated slab. Furthermore, let $\overrightarrow{m}$ be a unit normal to τ chosen so that $\overrightarrow{m}\cdot\overrightarrow{n} > 0$. (Note that $\overrightarrow{m}\cdot\overrightarrow{n} \neq 0$, as $\phi > 0$.) Let $\overrightarrow{u} = \overrightarrow{n} - (\alpha\overrightarrow{n}\cdot\overrightarrow{m})\overrightarrow{m}$ such that $0 < \alpha \leq 1$ and the slab π' with bounding planes π'_1 and π'_2 and unit normal $\overrightarrow{u}/|\overrightarrow{u}|$ is empty. First, we observe that $0 < |\overrightarrow{u}| < 1$. This follows from the fact that $|\overrightarrow{u}|^2 = \overrightarrow{u}\cdot\overrightarrow{u} = 1 - \alpha(2-\alpha)(\overrightarrow{m}\cdot\overrightarrow{n})^2$, $0 < \alpha(2-\alpha) \leq 1$ and $0 < \overrightarrow{m}\cdot\overrightarrow{n} = \cos\phi < 1$. Then,

$$\begin{aligned}
\mathrm{dist}(\pi'_1, \pi'_2) &= (p_{21} - p_{11})\cdot\overrightarrow{u}/|\overrightarrow{u}| \\
&= (p_{21} - p_{11})\cdot(\overrightarrow{n} - (\alpha\overrightarrow{n}\cdot\overrightarrow{m})\overrightarrow{m})/|\overrightarrow{u}| \\
&= (p_{21} - p_{11})\cdot\overrightarrow{n}/|\overrightarrow{u}| - (\alpha\overrightarrow{n}\cdot\overrightarrow{m})(p_{21} - p_{11})\cdot\overrightarrow{m}/|\overrightarrow{u}| \\
&= (p_{21} - p_{11})\cdot\overrightarrow{n}/|\overrightarrow{u}| > (p_{21} - p_{11})\cdot\overrightarrow{n} = \mathrm{dist}(\pi_1, \pi_2),
\end{aligned}$$

contradicting the optimality of π^*.

The remaining cases correspond to input sets where the points of $S_1 \cup S_2$ are not coplanar, and logically cover all possibilities not yet covered by cases (a) or (b). For algorithmic purposes it is useful to distinguish between inputs where the points of S_1 are collinear and those where they are not. Whenever (c) or (d) applies there is no rotation of the slab planes that preserves point incidences, so no additional information on the orientation of π_1 and π_2 can be derived.

It is not difficult to construct instances where each of the four cases occurs (see Figure 1). This shows that all cases are necessary and completes the proof.

As a consequence of the preceding lemma we can restrict our search to slabs C that satisfy one of the four conditions. We will denote by C_{11}, C_{21}, C_{31}, C_{22} the set of candidate slabs that satisfy the conditions of cases (a), (b), (c) and (d), respectively. Representatives from each set are shown in Figure 1.

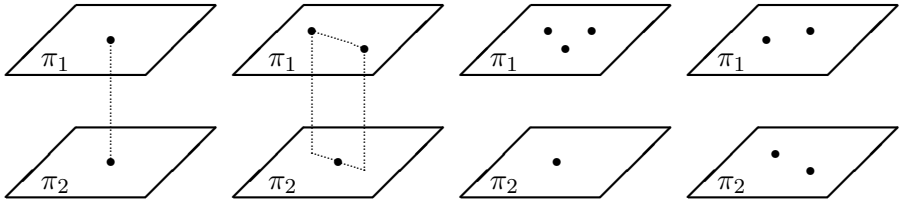


Fig. 1. Types of candidate slabs according to Lemma 1

3 Computing the Candidates

The optimal slab in C_{11} can be solved separately in $O(n^3)$ time by brute force. We describe how to compute optimal slabs in C_{21}, C_{31} and C_{22}.

Our approach is based on topological sweeps over the arrangement of planes corresponding to a dual representation of the points in S. We need to reinterpret the conditions (b), (c), and (d) of Lemma 1 in the dual space in order to find the solution using the arrangement. We use the transformation $\mathcal{D}$ which maps a point $p = (a, b, c)$ to the plane $\mathcal{D}(p) : z = ax + by - c$ in the dual space, and maps a non-vertical plane $\pi : z = mx + ny - d$ to the point $\mathcal{D}(\pi) = (m, n, d)$ in the dual space.

Since the dual transformation cannot handle vertical planes, we first discuss how to solve the special case in which the optimal plane is vertical.

Lemma 2. *The optimal vertical obnoxious plane can be computed in $O(n^2)$ time and $O(n)$ space.*

Proof. For a vertical plane π let $\ell(\pi)$ denote its intersection with the plane $z = 0$. Also, For a point $p \in S$, let p^* denote its orthogonal projection onto the plane $z = 0$. Note that, for any parallel vertical planes π_1 and π_2, $\text{dist}(\pi_1, \pi_2) = \text{dist}(\ell(\pi_1), \ell(\pi_2))$. Furthermore, a point $p \in \pi_i$ iff $p^* \in \ell(\pi)$. These facts, allow us to reduce the case of vertical slabs to the widest empty slab in 2-d. We build a set $S^* = \{p^*, p \in S\}$ and apply the algorithm of [10] to S^*. This algorithm runs in $O(n^2)$ time and $O(n)$ space. The equation of the optimal line for S^*, when interpreted in 3-d, is precisely the equation of the optimal vertical plane.

As a consequence of this lemma, we can restrict our attention to non-vertical slabs. Moreover, we assume that the points in S are in general position. In other words, we assume that, in dual space, every two planes intersect in a line, every three meet in a single point, and no four planes have a point in common.

Let H denote the set of planes $\{\pi_p = \mathcal{D}(p), p \in S\}$, and $\mathcal{A}(H)$, the arrangement of $\mathbb{R}^3$ induced by H. The properties of the duality transform can be used to characterize in $\mathcal{A}(H)$ the sets of slabs C_{21}, C_{31}, and C_{22}.

Let C be a slab with bounding planes π' and π''. The width of C can be computed using Observation 1. Since π' and π'' are parallel, the points $\mathcal{D}(\pi')$

and $\mathcal{D}(\pi'')$ lie on a vertical line in the dual space. Thus the slab C is represented in the dual space by the vertical segment $\mathcal{D}(C)$ with endpoints $\mathcal{D}(\pi')$ and $\mathcal{D}(\pi'')$. In fact, an empty slab in C_{31} corresponds to a vertical segment inside a cell of $\mathcal{A}(H)$ that connects a vertex and a face of that cell. Similarly, the empty slabs of C_{21} and C_{22} correspond to vertical segments inside cells of $\mathcal{A}(H)$ that connect an edge with a face, and an edge with an edge, respectively. By systematically examining these vertical segments we can report the overall widest empty slab. We now explain how to do this.

3.1 Finding the Solution in the Arrangement

In this section we describe a simple method, based on topological sweep in 3-d, to compute the optimal non-vertical slab in $O(n^3)$ time and $O(n^2)$ space. The idea is to sweep over $\mathcal{A}(H)$ while at any given time only storing a portion of it.

In dual space, for each cell of $\mathcal{A}(H)$ we examine all of the vertical segments that connect a vertex with a face, all of those that connect two edges and, by taking advantage of the orthogonality condition of Lemma 1(b), a selected subset of those that connect an edge with a face. To this end, we adapt the topological sweep algorithm of [2]. This algorithm requires $O(n^3)$ time and $O(n^2)$ working space when the planes of H are in general position.

We briefly review the mechanics of the topological sweep. The approach followed by [2] generalizes to 3-d the method proposed in [8] for sweeping an arrangements of lines in 2-d. Since $\mathcal{A}(H)$ may contain $\Theta(n^3)$ vertices and $\Theta(n^2)$ lines, the 3-d algorithm is optimal with respect to time complexity.

The idea is to sweep with a continuous unbounded surface that shares a point with each of the $O(n^2)$ lines of $\mathcal{A}(H)$. The *cut* is defined to be the set of segments or rays of $\mathcal{A}(H)$ intersected by the sweeping surface. Initially the surface is a plane perpendicular to the x-axis, and positioned to the left of the leftmost vertex of $\mathcal{A}(H)$. The sweep surface then advances from vertex to vertex. The transition of the surface from the left of a vertex to its right is called a *3-d elementary step*. Such a step consists of three 2-d steps, one on each of the three defining planes of the vertex.

The algorithm can perform an elementary step provided there exists at least one vertex with all three of the left-going edges in the current cut. Since this condition is always satisfied, the algorithm can perform elementary steps until all the vertices have been swept. To discover where in a cut an elementary step can be applied, a data structure based on the horizon tree [8] is used. This data structure stores information about the cells intersected by the sweep surface. The data structure requires $O(n^2 \log n)$ time for initialization and $O(1)$ amortized time per elementary step. Consequently, the overall sweep takes $O(n^3)$ time. The space complexity is $O(n^2)$ due to the use of a "local" data structure that requires $O(n)$ space for each plane of the arrangement.

In order to solve our problem, we perform a topological sweep of $\mathcal{A}(H)$. When leaving a cell c, we test every vertex-face, edge-edge, and edge-face pair of c in order to identify and compute the width of all pairs that are vertically aligned, i.e., all pairs that can be joined by a vertical segment interior to c. These pairs

correspond to candidates from C_{31}, C_{22} and C_{21}, respectively, associated with c. As described below, each candidate can be processed in $O(1)$ amortized time. While performing the sweep, we keep the vertices, edges and faces of all active cells. This can be done using $O(n^2)$ space as described in [2].

The details on how to process a candidate slab depend on its type. We now elaborate on this.

C_{31}: When leaving a cell c, we compute the width of each vertex-face pair associated with c and update the maximum every time a better candidate is found. In order to do this, for each vertex v of c we identify the face of c intersected by a vertical segment, interior to c, emanating from v. This is done by comparing the vertex against all faces of c. We then compute the width of the slab associated with this segment by using Observation 1.

C_{22}: The edge-edge pairs of C_{22} can be identified and reported as in the C_{31} case. We omit the details.

C_{21}: The width of the edge-face pairs of C_{21} can be computed as in C_{31}. Identifying the candidates, however, is more difficult. This is due to the fact that the number of vertical segments associated with an edge-face pair is not finite. Each such segment corresponds to an empty slab. Fortunately, the orthogonality condition of Lemma 1(b) can be used to identify the desired candidates as follows. Suppose that in dual space we find a vertical segment s connecting a point p_e on the edge e to a point p_f on the face f of a cell c. The parallel planes π_1 and π_2 in primal space that correspond to points p_e and p_f, respectively, have common normal vector $\overrightarrow{n_s} = (x(s), y(s), -1)$. Let p_{1i} and p_{1j} be the input points associated with the dual planes incident on e. and let p_{2k} be the input point associated with dual face f. The plane π passing through p_{1i}, p_{1j} and p_{2k} has normal vector $\overrightarrow{n} = (p_{1i} - p_{2k}) \times (p_{1j} - p_{2k})$. Edge e and face f determine a candidate slab if plane π is orthogonal to planes π_1 and π_2, i.e., if $\overrightarrow{n_s} \cdot \overrightarrow{n} = 0$. By parameterizing edge e in terms of its endpoints it is straightforward to determine a point on e that satisfies the orthogonality condition or to conclude that such a point does not exist. This computation takes $O(1)$ time.

The following lemma allows us to compute an upper bound on the total number of candidates in $C_{31} \cup C_{22} \cup C_{21}$ as well as a bound on the time required to identify those candidates.

Lemma 3. *[3] Let $\mathcal{A}(H)$ be the arrangement of a collection of n planes in R^3. For each cell c of $\mathcal{A}(H)$ let $f_i(c)$ denote the number of i-dimensional faces of the boundary of c, for $i = 0, 1, 2$, and let $f(c) = \Sigma_{i=0}^{2} f_i(c)$. Then $\Sigma f(c)^2 = O(n^3)$ where the sum extends over all cells of $\mathcal{A}(H)$.*

The result below is now a simple consequence of the previous discussion and the fact that the total number of vertex-face, edge-edge and edge-face pairs inside the cells of $\mathcal{A}(H)$ is bounded by $\Sigma f(c)^2$. Note, in particular, that Lemma 3 allows us to identify all candidate slabs for all cells in $O(n^3)$ time.

Theorem 1. *An obnoxious plane though a set of n points in $\mathbb{R}^3$ can be computed in $O(n^3)$ time and $O(n^2)$ space.*

Clearly, if degeneracies are present and if a topological sweep algorithm that handles them is not available, one can first construct $\mathcal{A}(H)$ explicitly. This should be done using a robust algorithm, such as the incremental solution coupled with simulation of simplicity, described in [7]. When doing this, the space complexity increases to $O(n^3)$ while the time complexity remains the same.

4 The Constrained Problems

In this section we consider constrained versions of the obnoxious plane problem where the optimal plane is required to pass through a fixed line or point.

The line-constrained version can be stated as follows. Given a set S of n points and a line ℓ in $\mathbb{R}^3$, compute a plane π passing through ℓ such that $\min_{p\in S} d(p,\pi)$ is maximal.

Without loss of generality, we assume that the line ℓ is the x-axis. We seek an optimal obnoxious plane π through this axis. Let π_α denote the plane whose normal $\overrightarrow{n}$ makes an angle α with the y-axis. Thus, we are looking for the value of $\alpha \in [0,\pi)$ such that $\min_{p\in S} d(p,\pi_\alpha)$ is maximal.

The proposed algorithm partitions the interval $[0,\pi)$ into subintervals such that all (rotated) planes in the same subinterval have the same point $p \in S$ as the nearest point. To compute the optimal value of α, it suffices to compute the lower envelope of n univariate functions $d(p,\pi_\alpha)$, $p \in S$.

The following result is crucial for computing the lower envelope efficiently.

Lemma 4. *Let p and q be two distinct points of S. Then, the functions $d(p,\pi_\alpha)$ and $d(q,\pi_\alpha)$ have at most two points of intersection.*

Proof. Let π be a plane passing trough the x-axis. The vector $\overrightarrow{n} = (0, \cos\alpha, \sin\alpha)$, $\alpha \in [0,\pi)$, is normal to the plane π_α. In other words, π_α has equation $\cos\alpha\ y + \sin\alpha\ z = 0$. Observe that for any two points $p = (p_1,p_2,p_3)$ and $q = (q_1,q_2,q_3)$ in S, the intersection of $d(p,\pi_\alpha)$ with $d(q,\pi_\alpha)$ satisfies $|p_2\cos\alpha + p_3\cos\alpha| = |q_2\cos\alpha + q_3\cos\alpha|$. Thus, the distance functions have common points for $\alpha = \arctan(\frac{q_2-p_2}{p_3-q_3})$ and $\alpha = \arctan(\frac{q_2+p_2}{p_3+q_3})$. Furthermore, the functions coincide when $p_2 = q_2 = 0$ and $|p_3| = |q_3|$. This proves the claim.

Let L_S be the lower envelope of the graphs of $d(p,\pi_\alpha)$, $p \in S$. Lemma 4 implies that the identifiers of the points corresponding to the edges of L_S, when traversing $L_\mathcal{P}$ from left to right, form a Davenport-Schinzel sequence of order two ([12]). Then, by divide-and-conquer, we can compute L_S in $O(n\log n)$ time. Furthermore, the number of intervals in the partition is in $O(n)$ ([12]). Thus, by traversing L_S, from left to right, we can identify the highest vertex, which corresponds to the optimal direction for π_α. This leads an $O(n\log n)$-time algorithm. A lower bound $\Omega(n\log n)$ for this problem can be obtained by reducing the *largest empty anchored cylinder* problem of [9] to our problem. Given n points on a plane Π, and an anchor point $O \in \Pi$, we consider the line l through O perpendicular to Π. The optimal obnoxious plane constrained to l solves the 2-d problem of [9]. Since any solution to the problem of [9] requires $\Omega(n\log n)$ time

under the algebraic computation tree model, our algorithm is optimal under this model. In summary, we have proven the result below.

Theorem 2. *The line-constrained obnoxious plane can be computed in optimal* $O(n \log n)$ *time and* $O(n)$ *space.*

The point-constrained problem can be stated as follows. Given a set S of n points and a point p_o, all in $\mathbb{R}^3$, compute a plane π_o through p_o such that $\varepsilon_o = \min_{p \in S} d(p, \pi_o)$ is maximal.

We can extend the approach described in the previous section by considering a finite collection of surfaces in 3-d space. To compute the lower envelope of the n bivariate functions we can use the divide-and-conquer deterministic approach of [1]. Thus, assuming an appropriate model of computation, we establish the following result.

Theorem 3. *The point-constrained obnoxious plane can be computed in* $O(n^{2+\varepsilon})$ *time and space.*

5 The Obnoxious Plane through Polyhedral Objects

Let $\mathcal{O}$ be a set of simple polyhedral objects in $\mathbb{R}^3$ with a total of n vertices. The **OPP** for $\mathcal{O}$ consists of finding a plane which maximizes the minimum distance to the objects. An empty slab C through $\mathcal{O}$ is an open region that intersects no objects from $\mathcal{O}$ and is enclosed by two parallel planes that intersect the convex hull of $\mathcal{O}$. Note that for a given $\mathcal{O}$ an empty slab may not exist. It is not difficult to prove that the bounding planes of a widest empty slab through $\mathcal{O}$ satisfy one of the four conditions of Lemma 1, except that the points p_{ij} are now vertices of polyhedra in $\mathcal{O}$.

Our method extend the approach of [11] to the three dimensional space. Let E be the set of edges of the polyhedra in $\mathcal{O}$. The dual representation of edge $e \in E$ with endpoints p and p' is the double-wedge $W(e)$ formed by planes $\mathcal{D}(p)$ and $\mathcal{D}(p')$ that does not contain the vertical plane through the line $\mathcal{D}(p) \cap \mathcal{D}(p')$. A plane π intersects e if and only if point $\mathcal{D}(\pi)$ lies inside $W(e)$.

Let $\mathcal{A}(H)$ be the arrangement of the n dual planes of the vertices of $\mathcal{O}$. (This arrangement is the same as the arrangement of the planes bounding the double-wedges $W(e)$ for $e \in E$.) Let π_1 and π_2 be two planes in the primal space. Planes π_1 and π_2 intersect the same edges of E (and therefore the same number of edges) if and only if $\mathcal{D}(\pi_1)$ and $\mathcal{D}(\pi_2)$ lie in the same cell of $\mathcal{A}(H)$. Let $count(c)$ denote the number of edges of E intersected by any plane whose dual lies inside cell c of $\mathcal{A}(H)$. When $count(c) = 0$, the points in c correspond to planes in the primal that do not intersect any edge of E. Consequently, an open vertical segment whose endpoints lie on the boundary of a cell c with $count(c) = 0$ is the dual of an empty slab through $\mathcal{O}$.

To find the widest empty slab through $\mathcal{O}$, we use the topological sweep algorithm described in Section 3.1, but consider only the cells c of $\mathcal{A}(H)$ for which $count(c) = 0$. To identify these cells, we adapt a technique of [2] to compute $count(c)$ in $O(1)$ time for each cell c of $\mathcal{A}(H)$. This computation is done when

a cell is first encountered during the sweep. At the start of the algorithm, we compute the count for each of the $O(n^2)$ cells cut by the initial topological plane. Since we have $O(n)$ edges, this takes $O(n)$ time per cell and $O(n^3)$ time altogether. Consider now the computation of *count*(c) during the sweep. Suppose that the sweep plane first encounters c at a vertex formed by the intersection of planes v_1^*, v_2^*, v_3^*, corresponding to vertices v_1, v_2, v_3 of $\mathcal{O}$, respectively. Let c' be the cell of $\mathcal{A}(H)$ left behind by the sweep plane when c is first encountered. To compute *count*(c) from *count*(c') we consider only the double-wedges that may change the count at c. Initially, *count*(c) is set to *count*(c'). Then, we increment the count for each double-wedge that contains c but not c', and decrement it for each double-wedge that contains c' but not c. The time to do this is proportional to the number of edges from E incident on v_1, v_2, or v_3. This number is at most nine, as the worst case occurs when the plane through v_1, v_2, v_3 does not contain any edges from E. Consequently, *count*(c) can be computed from *count*(c') in $O(1)$ time, and the result below follows.

Theorem 4. *An obnoxious plane through a set of polyhedral objects in $\mathbb{R}^3$ with a total of n vertices can be computed in $O(n^3)$ time and $O(n^2)$ space.*

6 Computing a Largest Empty Annulus

In [6] it is shown that given a set of n points S in $\mathbb{R}^2$, an empty annulus A (open region between two concentric circles) with largest width that partitions S into two subsets of points can be computed in $O(n^3 \log n)$ time an $O(n)$ space. We present an alternative algorithm to solve this problem in $O(n^3)$ time and $O(n^2)$ space.

Let us borrow the notation of [6]. Let $o(A)$ and $O(A)$ denote the inner and outer boundary of the circles defining A. Let $w(A)$, the width of A, be the positive difference between the radii of $O(A)$ and $o(A)$. An empty annulus of greatest width is a *syzygy* annulus if there are points p, q, with $p \in S \cap o(A)$ and $q \in S \cap O(A)$, and p is contained in the open segment whose endpoints are the center of the inner circle and q. As pointed out in [6], there always exist a largest empty annulus A such that (1) A is not a sizygy annulus and $|S \cap o(A)| \geq 2$ and $|S \cap O(A)| \geq 2$, or (2) A is a sizygy annulus and $|S \cap o(A)| \geq 2$ and $|S \cap O(A)| \geq 1$.

We first transform the set S from $\mathbb{R}^2$ to $\mathbb{R}^3$ by the well known paraboloid transformation $\mathcal{P} : p = (p_x, p_y) \rightarrow p^* = (p_x, p_y, p_x^2 + p_y^2)$. The point $p^\star$ is the vertical projection of point p onto the unit paraboloid $U : z = x^2 + y^2$ of $\mathbb{R}^3$. There is a one-to-one correspondence between circles in the original space and non-vertical planes in the transformes space.

It can be easily verified that the mapping $\mathcal{P}$ raises the annulus A of inner circumference $o(A) : x^2 + y^2 + ax + by + c = 0$ and outer circumference $O(A) : x^2 + y^2 + ax + by + d = 0$, with $c > d$, to the slab $A^\star$ bounded by the parallel planes $o(A)^\star : z + ax + by + c = 0$ and $O(A)^\star : z + ax + by + d = 0$. Reciprocally, any non vertical slab C bounded by planes $\pi : z + ax + by + c = 0$ and $\Pi : z + ax + by + d = 0$, with $c > d$, and both intersecting the unit paraboloid U, transforms to an annulus $C^\star$ whit inner and outer circumferences $\pi^\star : x^2 + y^2 +$

$ax + by + c = 0$ and $\Pi^\star : x^2 + y^2 + ax + by + d = 0$ respectively, and with width $w(C^\star) = 1/2\left[\sqrt{a^2 + b^2 - 4d} - \sqrt{a^2 + b^2 - 4c}\right]$. Observe also that a point p lies on (respectively inside, outside) a circle c if and only if the dual hyperplane $c^\star$ contains (respectively passes above, below) the dual point $p^\star$. Thus, the largest empty annulus problem in the plane reduces to the largest empty slab problem in the space. In fact, there are two cases to be considered. (1) The optimal slab corresponds to a non sizygy annulus (the candidate slabs can be determined adapting the C_{22} case) or (2) the optimal slab corresponds to a sizygy annulus (the candidate slabs can be determined adapting the C_{21} case). Consequently, we have the following result.

Theorem 5. *Given a set of n points in $\mathbb{R}^2$, a largest empty annulus can be computed in $O(n^3)$ time and $O(n^2)$ space.*

Acknowledgments. The first author was supported by Project MCyT: BFM2000-1052-C02-01. The second author was supported by the National Science Foundation under grant DMS-0107628. The third author was supported by grant TIC2001-2392-C03-01.

References

1. Agarwal P., Schwarzkopf O. and Sharir M.: The overlay of lower envelopes and its applications. Discrete and Computational Geometry **15** (1996) 1–13.
2. Anagnostou, E. G., Guibas, L. J., and Polimenis V. G.: Topological sweeping in three dimensions. Proceedings of the International Symposium on Algorithms (SIGAL). Lecture Notes in Computer Science **450** (1990) 310–317.
3. Aronov, B., Matousek, J. and Sharir M.: On the sum of Squares of cell Complexities in Hyperplane Arrangements. J. Combin. Theory Ser A **65** (1994) 311–321.
4. Chattopadhyay S. and Das P.: The k-dense corridor problems. Pattern Recogn. Lett. **11** (1990) 463–469.
5. Cheng. S.-W.: Widest empty L-shaped corridor. Inf. Proc. Lett. **58** (1996) 277–283.
6. Díaz-Báñez J.M., F. Hurtado F., Meijer H., Rappaport D. and Sellarès T.: The largest empty annulus problem. International Journal of Computational Geometry and Applications **13(4)** (2003) 317–325.
7. Edelsbrunner H.: *Algorithms in Combinatorial Geometry*. Springer-Verlag (1987).
8. Edelsbrunner H. and Guibas L.: Topologically sweeping an arrangement. Journal of Computer and System Sciences **38** (1989) 165–194.
9. Follert, F., Schömer, E., Sellen, J., Smid, M., Thiel, C.: Computing a largest empty anchored cylinder and related problems. International Journal of Computational Geometry and Applications **7**, (1997) 563–580.
10. Houle, M., Maciel A.: Finding the widest empty corridor through a set of points, In G.T. Toussaint, ed., *Snapshots of computational and discrete geometry*, 210-213. TR SOCS-88.11, dept of Computer Science, McGill University, Canada, 1988.
11. Janardan R., Preparata F.P.: Widest-corridor problems. Nordic Journal of Computing **1** (1994) 231–245.
12. Sharir, M. and P.K. Agarwal: Davenport-Schinzel Sequences and Their Geometric Applications. Cambridge University Press, 1995.
13. Shin C., Shin S. Y., Chwa K.: The widest k-dense corridor problems. Information Processing Letters **68(1)** (1998) 25–31.

3D-Color-Structure-Code – A New Non-plainness Island Hierarchy

Patrick Sturm

Universität Koblenz-Landau, Institute of Computational Visualistics,
Universitätsstraße 1, 56070 Koblenz, Germany
sturm@uni-koblenz.de

Abstract. The Color Structure Code (CSC) [5] is a very fast and robust region growing technique for segmentation of color or gray-value images. It is based on a hierarchical hexagonal grid structure of the 2d space that fulfills several interesting topological properties. It is known that not all of these properties can be fulfilled together in 3d space. Here we introduce a new 3d hierarchical grid structure that fulfills the most interesting properties. A 3d CSC-segmentation based on this grid structure has been implemented.

1 Introduction

Image segmentation is an important step in image analysis. It divides an image into possibly large, pairwise disjoint segments [3]. Segments are defined as *spatial connected* pixel sets (pixel = location + color) that fulfill some homogeneity criterion. Generally, segments could be considered to be homogeneous in gray value, color or texture. Segmentation is used to get an abstract, symbolic representation of an image. The quality of an image analysis depends often strongly on the quality of the segmentation result. Nowadays 3-dimensional images play an important role especially in medical imaging. They are generated, e.g., by diagnostic methods like Computer Tomography (CT), Magnet Resonance Tomography (MRT) and Positron Emission Tomography (PET). Thus, it is a valuable task to generalize the very successful 2d CSC-segmentation technique ([4], [5]) to 3d images. However, this approach leads to some surprising difficulties.

2 The Hexagonal Island Hierarchy

The CSC follows a hierarchical region growing on a special hierarchical hexagonal topology that was firstly introduced by Hartmann [2]. This hierarchical topology (see fig. 1a) is formed by so-called *islands* of different levels. One island of level 0 (denoted by I^0) consists of seven pixels (one center pixel and its 6 neighbors) in the hexagonal topology. The partition of the image is organized in such a way that the islands are overlapping (each second pixel of each second row is a center of an island of level 0). One island of level n+1 (denoted by I^{n+1}) consists of seven

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 109–116, 2004.
© Springer-Verlag Berlin Heidelberg 2004

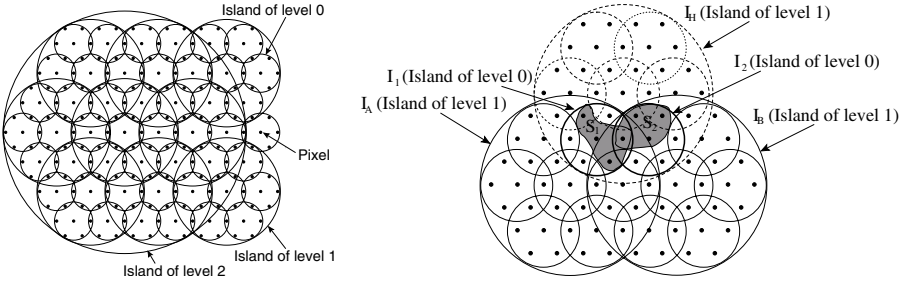


Fig. 1. a The Hexagonal Island Hierarchy**. b** Island I_H covers both islands I_1 and I_2.

overlapping islands of level n s. fig. 1a). Repeating this until one island covers the whole image the number of islands decreases from level to level by factor 4.

A simple neighborhood relation can be defined on islands of level n: Two islands I_1^n and I_2^n are neighbors iff I_1^n and I_2^n overlap each other, i.e. $I_1^n \cap I_2^n \neq \emptyset$. Each island I^{n+1} consists of seven islands of level n (a center island I_0^n and its 6 neighbors $I_1^n, ..., I_6^n$). These seven islands $I_0^n, ..., I_6^n$ are called the *sub-islands* of I^{n+1}. I^{n+1} is called the *parent island* of $I_0^n, ..., I_6^n$. In the hexagonal island hierarchy all islands of level n may have up to *two* different parent islands: A center of an island of level n is never part of an overlapping between two different islands of level n, i.e. center islands have exactly one parent island. All other islands are part of an overlapping between two different islands of level n. Therefore they have exactly two parent islands.

3 Properties of Island Hierarchies

The hexagonal island hierarchy is a very special hierarchical topology. One can imagine a lot of other hierarchies with different properties, e.g. the overlapping structure of the islands may be more complex compared to the overlapping structure of the islands in the hierarchical hexagonal topology. The following list contains general properties of island hierarchy. All of these properties are all fulfilled by the hexagonal island hierarchy.

1. **Homogeneity:** All islands of level $n+1$ comprise the same number of *sub-islands*.
2. **k-Neighborhood:** Each island I^n of level n overlaps with exactly k different neighbor islands of level n.
3. **Plainness:** Two islands of level $n+1$ overlap each other in at most one island of level n.
4. **Reducibility:** The number of islands are reduced from level to level by factor 2^d where d is the dimension of the under lying grid.
5. **Connectivity:** The sub-islands $I_0^n, ..., I_k^n$ of each island I^{n+1} are connected pairwise within I^{n+1}. Two sub-islands I_i^n, I_j^n of I^{n+1} are connected within I^{n+1}, iff there is a path of overlapping sub-islands between I_i^n, I_j^n in I^{n+1}, i.e. $\exists x_0, ..., x_m \in \{0, ..., k\} : x_0 = i \wedge x_m = j \wedge \forall 0 \leq s < m : I_{x_s}^n \cap I_{x_{s+1}}^n \neq \emptyset$.

Center Connectivity: Each island of level $n+1$ consists of a center island of level n and its (overlapping) neighbor islands. Center Connectivity is a stronger condition than *connectivity*.

6. ***Strong* (resp. *weak*) Saturation:** All sub-islands (except the center island) of an island of level $n+1$ are sub-islands of *exactly* (resp. *at least*) two different islands of level $n+1$. Each center island has *exactly* (resp. at least) one parent island.
7. **Coverability:** Each island of level n (except the topmost island) is a sub-island of at least one island of level $n+1$.
8. **Density:** Two neighboring islands I_1^n and I_2^n are always sub-islands of a common island I^{n+1}, i.e. $I_1^n \in I^{n+1} \wedge I_2^n \in I^{n+1}$.

There are two interesting propositions for island hierarchies:

Proposition 1: Given a d-dimensional island hierarchy $\mathcal{H}$ that fulfills the *homogeneity*, *plainness*, *reducibility* and *strong saturation* properties. All islands of level $n+1$ in $\mathcal{H}$ must have exactly $k = 2^{d+1}-1$ sub-islands of level n.

Proof: Let be m the number of islands of level $n+1$ in $\mathcal{H}$. Due to the *homogeneity* property each island of level $n+1$ has k sub-islands. To get the number of islands of level n in $\mathcal{H}$ – denoted by $C(\mathcal{H})$ – it is not enough just to multiply m with k. Due to the *plainness* and *strong saturation* properties of $\mathcal{H}$ such islands that are not center islands would be counted twice. If we add the number of center islands ($= m$) to the term *all* islands of level n will be counted twice: $2 \cdot C(\mathcal{H}) = m \cdot k + m$. Due to the reducibility of $\mathcal{H}$ $C(\mathcal{H})$ can be expressed also by $C(\mathcal{H}) = m \cdot 2^d$. Both expressions are equivalent iff $k = 2^{d+1}-1$.

From Proposition 1 follows directly that an d-dimensional island hierarchy whose islands have less or more than $k = 2^{d+1}-1$ sub-islands cannot fulfill all 8 properties of the hierarchical hexagonal topology.

Proposition 2: If a d-dimensional island hierarchy $\mathcal{H}$ fulfills the *homogeneity*, *reducibility* and *weak saturation* properties and if the islands of $\mathcal{H}$ consists of $k < 2^{d+1}-1$ sub-structures the *coverability* property cannot be fulfilled for $\mathcal{H}$.

Proof: Let be m the number of islands of level n. Due to the *reducibility* property of $\mathcal{H}$ the number of islands of level $n+1$ is given by the term $m/2^d$. Now let be $I_0^n, ..., I_{k-1}^n$ the k sub-islands of an island I^{n+1}. p_i denotes the number of parents of I_i^n. Without loss of generality I_0^n should be the island center and thus has at least one parent ($p_0 \geq 1$). Let be $C^n(\mathcal{H})$ the number of islands of level n that have at least one parent and thus fulfill the coverability property. It is not sufficient just to multiply the number of islands of level $n+1$ with k to obtain $C^n(\mathcal{H})$. Some islands would be counted several times since all I_i^n, $1 \leq i < k$ are sub-islands of at least two different islands (Weak Saturation, $p_i \geq 2$). Therefore we must not count each island fully: $C^n(\mathcal{H}) = \frac{m}{2^d} \cdot (1/p_0 + 1/p_1 + ... + 1/p_{k-1})$. Due to the condition $p_0 \geq 1 \wedge \forall 0 < i \leq k-1 : p_i \geq 2$ an upper limit for $C^n(\mathcal{H})$ is given by $\hat{C}^n(\mathcal{H}) := \frac{m}{2^d} \cdot (1 + (k-1) \cdot \frac{1}{2})$. For all $k < 2^{d+1}-1$ the inequality $C^n(\mathcal{H}) \leq \hat{C}^n(\mathcal{H}) < m$ is valid. Then there are at least $m - \hat{C}^n(\mathcal{H}) > 0$ islands of level n that have no parents. Thus the coverability property is not fulfilled for $\mathcal{H}$ if $k < 2^{d+1}-1$.

4 The Color Structure Code

The CSC is a fast and very robust region growing technique that depends heavily on the hexagonal island hierarchy. In a first (and trivial) step all local segments within a single island of level 0 are detected, independently for all those islands of level 0. In step n+1 we assume that all segments in each island of level n are already detected. For an island I^{n+1} of level n+1 consisting of seven sub-islands $I_0^n, ..., I_6^n$ of level n, one iteratively grows all segments S_1^n, S_2^n in two sub-islands I_i^n and I_j^n into a new segment of level n+1, if S_1^n and S_2^n are similar (in color or grey-value) and they overlap in a common sub-segment $S_{1,2}^{n-1}$ of level n–1. Details can be found in [5]. This concept works very well due to the nice properties of the hexagonal island structure.

The four most important properties for the segmentation task are the *weak saturation*, *coverability*, *connectivity* and *density* properties. The *weak saturation* is necessary, as otherwise, some segments of level n might not become connected in level n+1. The *coverability* and *density* properties have a more special meaning: Let be I^n an island without parents (violation of the coverability property). All segments that are detected within I^n cannot become sub-segments of larger segments of level n+1 because I^n does not have any parent island, i.e. these segments may not grow any further. In particular pixels without any parent island cannot become part of a segment at all. We call islands of level n (resp. pixels) without a parent island *holes*. An example of a 3d island hierarchy that violates the coverability property is presented in the next section. The violation of the density property may produce different but overlapping segments: The figure 1b) shows two color similar segments S_1 and S_2 that share a common pixel. S_1 was detected in level 0 island I_1 and S_2 in level 0 island I_2. Because I_1 and I_2 are both sub-islands of level 1 island I_H the segments S_1 and S_2 can only be linked to a new segment within I_H. S_1 and S_2 would not be linked together if I_H would be missing (violation of the density property). The *connectivity* property is important because the CSC should detect spatially connected segments. Therefore the sub-islands of each island in the hierarchy should be also connected.

The essential operation of the CSC segmentation method is to merge overlapping and color similar segments of level n within islands of level n+1 to larger segments of level n+1. This ability is already ensured by the *weak saturation*, *coverability*, *connectivity* and *density* properties. The *homogeneity*, *k-neighborhood*, *plainness*, *reducibility*, *center connectivity* and *strong saturation* properties leads further to a simple design of the CSC segmentation method. But they are not important for a working CSC.

5 The Sphere-Island-Hierarchy

In a first approach we try to find generalization of the hexagonal island hierarchy in 3d. Therefore we use the most dense sphere packing (abbr.: MDSP see fig. 2) as the underlying grid of the 3d island hierarchy. The MDSP [1] is the analogy of the hexagonal grid in 3d: The distance between two adjacent spheres (neighbors)

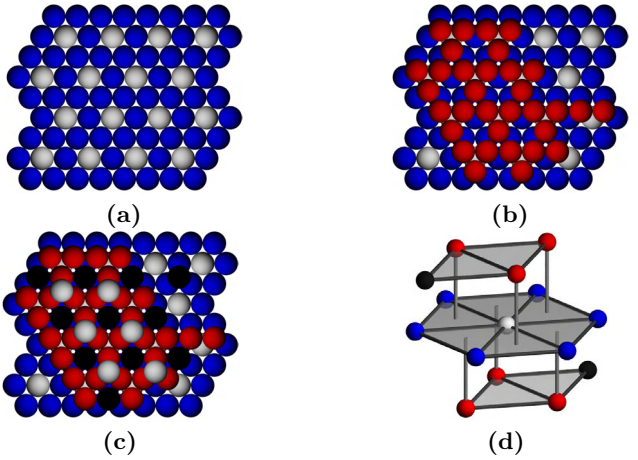


Fig. 2. a First layer of the MDSP. White spheres: Island centers. Gray spheres: Neighbors of island centers**. b** Second layer of the MDSP. Black spheres: Holes **. c** Third layer of the MDSP**. d** 3d island of level 0.

is always the same. Each sphere touches exactly 12 other spheres – 6 neighbors in the same layer and 3 neighbors each in the layer above and below. By increasing each sphere we will get the desired overlapping property. Each neighbor should overlap. We define in a first step a 3d island of level 0 as a set of 13 spheres – one center sphere and its 12 neighbors. Each second sphere of each second row of each second layer has to be a center sphere to ensure that two 3d islands of level 0 overlaps in at most one common sphere (Plainness) and that each sphere except the center sphere is covered by exactly two different islands (Strong Saturation). Islands of level $n+1$ consists of a center island of level n and its 12 neighbor islands. We call this kind of island hierarchy *Sphere-13* (abbr.: S_{13}). As one may see in fig. 2 there are some (black) spheres that are not a neighbor of any center sphere. Such spheres (holes) are not covered by any 3d island of level 0 (violation of the coverability property).

As we know from proposition 1 islands of a 3d island hierarchy that fulfill the *homogeneity*, *reducibility*, *strong saturation* and *coverability* properties must have exactly $k = 2^4 - 1 = 15$ sub-structures. Thus a 3d island of level 0 must consist of 15 spheres – a center sphere, its 12 neighbors and two additional holes that lie close to the island center (s. fig. 2d). Now each 3d island of level 0 overlaps with exactly 14 other islands of level 0. Generally a 3d island of level $n+1$ consists of one center island of level n and its 14 neighbors. Also a 3d island of level $n+1$ overlaps with exactly 14 other islands of level $n+1$. The center connectivity is fulfilled for all islands of level $n > 0$ but for islands of level 0. The center of an island of level 0 is never adjacent to a hole. From level to level the number of islands decreases by factor 8 (Reducibility). We call this modified S_{13} island hierarchy S_{15}. The island hierarchy S_{15} fulfills all properties but the density property. As this property is essential for the segmentation task S_{15} is not suitable. Nevertheless it is possible to use S_{15} for segmentation. For this purpose the segmentation algorithm has to be modified. Details can be found in [7].

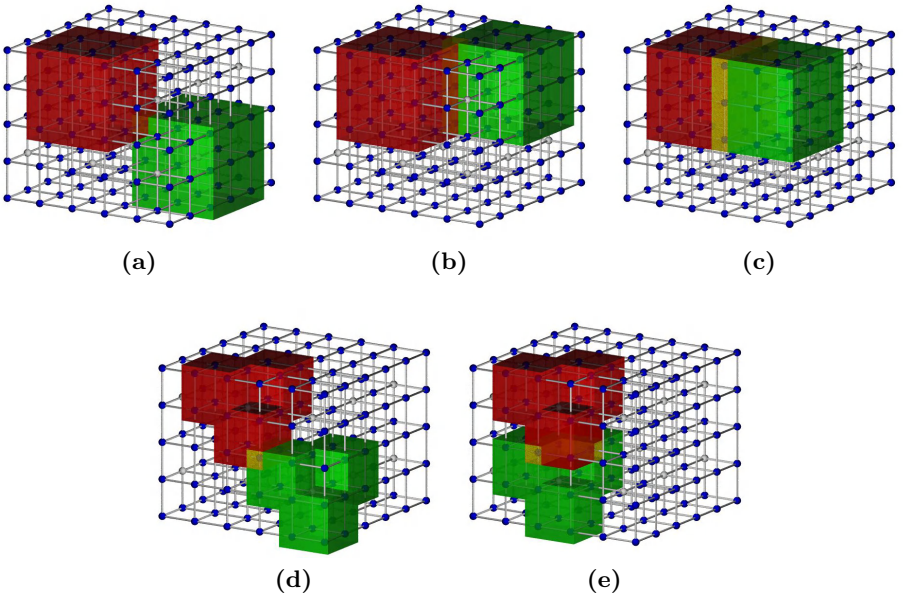


Fig. 3. Two C_{27} islands of level 0 overlap in a common **(a)** corner, **(b)** edge, **(c)** face. Two C_{19} islands of level 0 overlap in **(d)** 2 common voxels or **(e)** three common voxels.

6 Non-plainness Island Hierarchies

As the density property is important for an island hierarchy used by the CSC we are looking for a proper island hierarchy that fulfills at least the *connectivity*, the *coverability*, the *weak saturation* and the *density* properties.

One suitable island hierarchy follows the orthogonal topology. An island of level 0 consists of 27 voxels in the orthogonal topology – a center voxel and its 26 neighbors. The position of the islands of level 0 are distributed in such a way that each island of level 0 overlaps with 26 other islands of level 0 (Each second voxel of each second row of each second layer is a center voxel). We may consider these islands as *macro voxels* of an orthogonal grid. Thus all hierarchy levels can be build in the same way: An island of level n+1 consists of a center island of level n and its 26 overlapping neighbor islands. We say that this (orthogonal) island hierarchy is of type Cube-27 (abbr.: C_{27}) because all islands have 27 sub-islands and all islands looks like cubes (see fig. 3).

The island hierarchy C_{27} fulfills all properties but the *plainness* and *strong saturation* properties (see table 1). The overlapping structure of its islands is more complex compared to that of the sphere-island-hierarchy:

1. Two islands of level n+1 may overlap in 1, 3 or 9 common islands of level n (see fig. 3).
2. Each island may have 1, 2, 4 or 8 parent islands.

Table 1. Comparison of Island Hierarchies

Island Hierarchy	S_{13}	S_{15}	C_{19}	C_{27}
Island Size	13	15	19	27
Size(s) of Overlapping Areas	1	1	2 or 3	1, 3 or 9
Max. Number of Fathers	0, 1 or 2	1 or 2	1, 2 or 3	1, 2, 4 or 8
Center Connectivity	✓	✓	–	✓
Connectivity	✓	✓	✓	✓
Homogeneity	✓	✓	✓	✓
Plainness	✓	✓	–	–
Coverability (!)	–	✓	✓	✓
Saturation (!)	***Strong***	***Strong***	***Weak***	***Weak***
Density (!)	–	–	✓	✓

Overlapping regions may be connected in a lot of different ways within two overlapping islands due to their possibly large overlapping area. This leads to a less efficient segmentation algorithm because many connections have to be checked. Further the islands are very large. The processing of large islands needs much more computation time during the CSC-segmentation than the processing of small islands.

From a practical point of view the island hierarchy of type C_{27} is therefore not suitable for segmentation. Instead of using this island hierarchy for segmentation we use it as a starting point to find one with smaller islands. Therefore we iterate over all possible island hierarchies included in C_{27} and test whether they fulfill all three demanded properties or not. This is done by a program that runs on an Intel Pentium IV with 2.4GHz several days. The computer replaces all islands of type C_{27} by smaller islands of type $C_x \subset C_{27}$ where x denotes the number of sub-islands. To reduce the number of possibilities we search just for islands with $x \in \{15, ..., 23\}$ sub-islands. As we know x must be at least 15. Otherwise the coverability property is not fulfilled for the resulting island hierarchy. The value for the upper limit of x has no special meaning. But x should not to be too large, because large islands will increase the computation time of the 3D-CSC. Therefore we are looking for hierarchies with small islands. Solutions to this problem was found for $x = 19$ and for $x = 23$. As we are looking just for island hierarchies with small islands we reject the solutions of $x = 23$. It turns out that all hierarchies of type C_{19} are just rotated or mirrored versions of a common prototype island hierarchy. So the island hierarchy of type C_{19} is unique. Islands of type C_{19} can be imagined as three overlapping cubes with side length 2. The three cubes overlap pairwise in two common sub-islands. There are only two different overlapping types between two islands (see figure 3d-e): Two islands may overlap in 2 or 3 common sub-islands. An island may have only 2 or 3 parent islands. Compared to 8 parents of islands of type C_{27} this is not a lot. Now which properties are fulfilled for this hierarchy and which are not? The homogeneity, coverability and weak saturation are fulfilled. This was tested by the computer. The strong saturation and the plainness are obviously violated. What's about the center connectivity? Each island of type C_{19} overlaps only

with 12 other islands of the same level and not with 18. But an island consists of exactly 19 sub-islands. This means that each island of level $n+1$ do not consist just of a center island and its 12 neighbors. Thus the *center connectivity property* cannot be valid. But as stated before this property is not essential for a working CSC algorithm. Instead the *connectivity* property is valid for C_{19}.

The island hierarchy of type C_{19} is not as complex as the hierarchy of type C_{27}. It has small islands and the overlapping structure between them is not too complex. Thus this hierarchy is a real alternative to the sphere-island-hierarchy.

7 Outlook

Table 1 shows which properties are fulfilled by the island hierarchies S_{13}, S_{15}, C_{19} and C_{27} and which are not. For none of these four hierarchies all eight properties are fulfilled together. As there is a nice hierarchical island topology in 2d – the hexagonal island hierarchy – it is not supposable to find a hierarchy in $3d$ that fulfills all properties of the hexagonal island hierarchy. We have shown that a 3d island hierarchy could not fulfill all eight properties of the hexagonal island hierarchy if its islands have less or more than 15 sub-islands (see Proposition 1). But what if the number is equal to 15? We are working at a proof to show that there is no island hierarchy that fulfills all eight properties in 3d. There is another interesting open question: We are trying to drop only the property of homogeneity. Is it possible to find two different types of islands covering the 3d space in an overlapping way fulfilling all further properties? The problem is here to find a proper inductive rule for defining islands of higher levels than 0.

References

1. J.H. Conway, H.J.A. Sloane. Sphere Packings, Lattices and Groups. Third Edition. Springer, 1998.
2. G. Hartmann. Recognition of Hierarchically Encoded Images by Technical and Biological Systems. In: Biological Cybernetics, 57:73–84, 1987.
3. S. L. Horowitz, T. Pavlidis. Picture Segmentation by a Traversal Algorithm. Journal of the ACM, 23:368–388, 1976.
4. L. Priese, V. Rehrmann. A Fast Hybrid Color Segmentation Method. In: S.J. Pöppl and H.Handels, editors, Mustererkennung 1993, pages 297–304. Springer, 1993. 15. DAGM-Symposium, Lübeck, 27.-29.Sept. 1993.
5. V. Rehrmann, L. Priese. Fast and Robust Segmentation of Natural Color Scenes. 3rd Asian Conference on Computer Vision, Hongkong, 8–10th January 1998.
6. P. Sturm, L. Priese. Properties of a Three-Dimensional Island Hierarchy for Segmentation of 3D Images with the Color Structure. In: Luc van Gool, editor, Pattern Recognition, pages 274-281. Springer Verlag, 2002. 24th DAGM-Conference, Zürich, 16–18th September 2002.
7. P. Sturm, L. Priese. 3D-Color-Structure-Code. A Hierarchical Region Growing Method for Segmentation of 3D-Images. In: J.Bigun and T.Gustavson (Eds.)., Image Analysis, pages 603-608. Springer Verlag, 2003. SCIA 2003, Halmstadt, June/July 2003.

Quadratic-Time Linear-Space Algorithms for Generating Orthogonal Polygons with a Given Number of Vertices*

Ana Paula Tomás[1] and António Leslie Bajuelos[2]

[1] DCC-FC & LIACC, University of Porto, Portugal
apt@ncc.up.pt
[2] Dept. of Mathematics & CEOC - Center for Research in Optimization and Control, University of Aveiro, Portugal
leslie@mat.ua.pt

Abstract. We propose INFLATE-PASTE – a new technique for generating orthogonal polygons with a given number of vertices from a unit square based on gluing rectangles. It is dual to INFLATE-CUT – a technique we introduced in [12] that works by cutting rectangles.

1 Introduction

To test and evaluate geometric algorithms we may need to construct samples of random geometric objects. The main motivation for our work was the experimental evaluation of the algorithm described in [11]. In addition, the generation of random geometric objects raises interesting theoretical questions.

In the sequel, polygon stands for *simple polygon without holes* and sometimes it refers to a polygon together with its interior. P denotes a polygon and r the number of reflex vertices. A polygon is *orthogonal* if its edges meet at right angles. As usual, H and V are abbreviations of horizontal and vertical, respectively, e.g., H-edge, V-edge, H-ray and so forth. For every n-vertex orthogonal polygon (n-*ogon*, for short), $n = 2r + 4$, e.g. [7]. Generic orthogonal polygons may be obtained from a particular kind of orthogonal polygons, that we call *grid orthogonal polygons* (see Fig. 1). A *grid n-ogon* is any n-ogon in general position defined in a $\frac{n}{2} \times \frac{n}{2}$ square grid. P is in general position iff it has no collinear edges. We assume the grid is defined by the H-lines $y = 1, \ldots, y = \frac{n}{2}$ and the V-lines $x = 1, \ldots, x = \frac{n}{2}$ and that its northwest corner is (1,1). Every grid n-ogon has exactly one edge in every line of the grid. Each n-ogon which is not in general position may be mapped to an n-ogon in general position by ϵ-perturbations, for a sufficiently small constant $\epsilon > 0$. Hence, we may restrict generation to n-ogons in general position. Each n-ogon in general position is mapped to a unique grid n-ogon through top-to-bottom and left-to-right sweeping. Reciprocally, given a grid n-ogon we may create an n-ogon that is an instance of its class by randomly spacing the grid lines in such a way that their relative order is kept.

* Partially funded by LIACC through *Programa de Financiamento Plurianual, Fundação para a Ciência e Tecnologia (FCT)* and *Programa POSI*, and by CEOC (Univ. of Aveiro) through *Programa POCTI, FCT, co-financed by EC fund FEDER.*

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 117–126, 2004.
© Springer-Verlag Berlin Heidelberg 2004

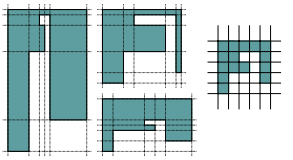


Fig. 1. Three 12-ogons mapped to the same grid 12-ogon.

The Paper's Contribution. We propose two methods that generate grid n-ogons in polynomial time: Inflate-Cut and Inflate-Paste. The former has been published in [12]. There we mention two programs for generating random orthogonal polygons, one by O'Rourke (developed for the evaluation of [8]) and another by Filgueiras[1]. O'Rourke's program constructs such a polygon by gluing together a given number of *cells* (i.e., unit squares) in a board, starting from a seed cell. The cells are chosen in a random way using heuristics. Filgueiras' method shares a similar idea though it glues rectangles of larger areas and allows them to overlap. Neither of these methods allows to control the final number of vertices of P. The major idea in Inflate-Paste is also to glue rectangles. Nevertheless, it strongly restricts the positions where rectangles may be glued. In this way, not only the algorithm becomes simpler and elegant, but also controls the final number of vertices and guarantees that P is in general position. The Inflate transformation is crucial. Inflate-Paste may be implemented so as to run in quadratic-time in the worst-case using linear-space in n. For the Inflate-Cut method we had the same space complexity, but could only guarantee average quadratic-time complexity, because Cut may fail. In addition, Inflate-Paste allows to understand the combinatorial structure of orthogonal polygons much better [2].

In the next section we describe the Inflate-Paste transformation and recall Inflate-Cut. In Sect. 3 we give a formal proof that both these techniques are complete. Finally, Sect. 4 is devoted to implementation and complexity issues.

2 Inflate, Cut, and Paste Transformations

Let $v_i = (x_i, y_i)$, for $i = 1, \ldots, n$, be the vertices of a grid n-ogon P, in CCW order.

2.1 Inflate

Inflate takes a grid n-ogon P and a pair of integers (p, q) with $p, q \in [0, \frac{n}{2}]$, and yields a new n-vertex orthogonal polygon $\tilde{P}$ with vertices $\tilde{v}_i = (\tilde{x}_i, \tilde{y}_i)$ given by $\tilde{x}_i = x_i$ if $x_i \leq p$ and $\tilde{x}_i = x_i + 1$ if $x_i > p$, and $\tilde{y}_i = y_i$ if $y_i \leq q$ and $\tilde{y}_i = y_i + 1$ if $y_i > q$, for $i = 1, \ldots, n$. Inflate augments the grid creating two free lines, namely $x = p + 1$ and $y = q + 1$.

[1] Personal communication, DCC-LIACC, 2003.

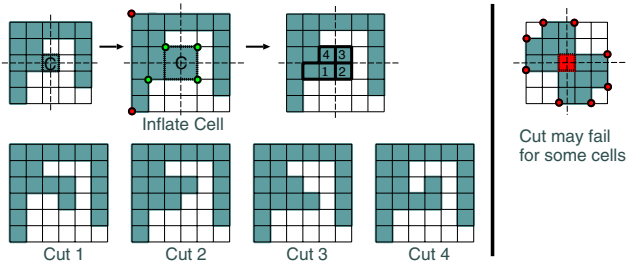


Fig. 2. The INFLATE-CUT transformation. The two rectangles defined by the center of C and the vertices of the leftmost V-edge $((1,1),(1,7))$ cannot be cut and so there remain the four possibilities shown. On the right we see a situation where CUT fails.

2.2 Inflate-Cut

Fig. 2 llustrates this technique. Let C be a unit cell in the interior of P, with center c and northwest vertex (p,q). When we apply INFLATE using (p,q), c is mapped to $\tilde{c} = (p+1, q+1)$, that is the center of inflated C. The goal of CUT is to introduce $\tilde{c}$ as a reflex vertex of the polygon. To do that, it cuts one rectangle (defined by $\tilde{c}$ and a vertex $\tilde{v}_m$ belonging to one of the four edges shot by the H- and V-rays that emanate from $\tilde{c}$). We allow such a rectangle to be cut iff it contains no vertex of $\tilde{P}$ except $\tilde{v}_m$. If no rectangle may be cut, we say that CUT fails for C.

So, suppose that $\tilde{s}$ is the point where one of these rays first intersects the boundary of $\tilde{P}$, that $\tilde{v}_m$ is one of the two vertices on the edge of $\tilde{P}$ that contains $\tilde{s}$ and that the rectangle defined by $\tilde{c}$ and $\tilde{v}_m$ may be cut. CUT cuts this rectangle from $\tilde{P}$ and replaces $\tilde{v}_m$ by $\tilde{s}$, $\tilde{c}$, $\tilde{s}'$ if this sequence is in CCW order (or $\tilde{s}'$, $\tilde{c}$, $\tilde{s}$, otherwise), with $\tilde{s}' = \tilde{c} + (\tilde{v}_m - \tilde{s})$. We may conclude that $\tilde{s}$, $\tilde{c}$, $\tilde{s}'$ is in CCW order iff $\tilde{s}$ belongs to the edge $\overline{\tilde{v}_{m-1}\tilde{v}_m}$ and in CW order iff it belongs to $\overline{\tilde{v}_m\tilde{v}_{m+1}}$. CUT always removes a single vertex of the grid ogon and introduces three new ones. CUT never fails if C has an edge that is part of an edge of P. Hence, the INFLATE-CUT transformation may be always applied to any P.

2.3 Inflate-Paste

We first imagine the grid n-ogon merged in a $(\frac{n}{2}+2)\times(\frac{n}{2}+2)$ square grid, with the top, bottom, leftmost and rightmost grid lines free. The top line is $x = 0$ and the leftmost one $y = 0$, so that $(0,0)$ is now the northwest corner of this extended grid. Let $e_{\mathrm{H}}(v_i)$ represent the H-edge of P to which v_i belongs.

Definition 1. *Given a grid n-ogon P merged into a $(\frac{n}{2}+2)\times(\frac{n}{2}+2)$ square grid and a convex vertex v_i of P, the* free staircase neighbourhood of v_i, *denoted by FSN(v_i), is the largest staircase polygon in this grid that has v_i as vertex, does not intersect the interior of P and its base edge contains $e_{\mathrm{H}}(v_i)$ (see Fig. 3).*

FSN(v_i) is the intersection of a particular quadrant (with origin at v_i) with the polygon formed by the external points that are rectangularly visible from v_i.

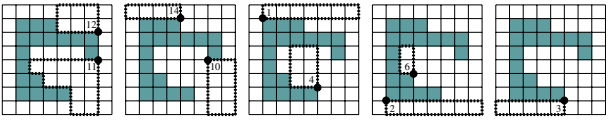


Fig. 3. A grid n-ogon merged into a $(\frac{n}{2}+2)\times(\frac{n}{2}+2)$ square grid and the free staircase neighbourhood for each of its convex vertices.

This quadrant is determined by $e_{\mathrm{H}}(v_i)$ and a V-ray emanating from v_i to the exterior of P. So, FSN(v_i) may be computed in linear time by adapting Lee's algorithm [4,5] or a sweep based method given by Overmars and Wood in [9]. We say that two points a and b are rectangularly visible if the axes-aligned rectangle that has a and b as opposite corners does not intersect the interior of P.

To transform P by Inflate-Paste (see Fig. 4) we first take a convex vertex v_i of P, select a cell C in FSN(v_i) and apply Inflate using the nortwest corner (p, q) of C. As before, the center of C is mapped to $\tilde{c} = (p + 1, q + 1)$, that will now be a convex vertex of the new polygon. Paste glues the rectangle defined by $\tilde{v}_i$ and $\tilde{c}$ to $\tilde{P}$, augmenting the number of vertices by two. If $e_{\mathrm{H}}(v_i) \equiv \overline{v_i v_{i+1}}$ then Paste removes $\tilde{v}_i = (\tilde{x}_i, \tilde{y}_i)$ and inserts the chain $(\tilde{x}_i, q + 1)$, $\tilde{c}$, $(p + 1, \tilde{y}_i)$ in its place. If $e_{\mathrm{H}}(v_i) \equiv \overline{v_{i-1} v_i}$, Paste replaces $\tilde{v}_i$ by the chain $(p + 1, \tilde{y}_i)$, $\tilde{c}$, $(\tilde{x}_i, q + 1)$. Clearly, Paste never fails, in contrast to Cut, because the interior of FSN(v_i) is nonempty, for every convex vertex v_i of P.

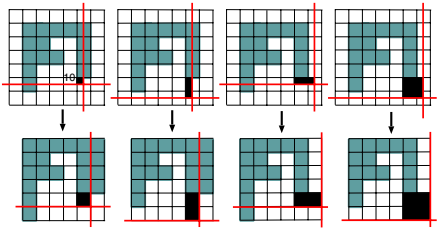


Fig. 4. At the bottom we see the four grid 14-ogons that may result when Inflate-Paste is applied to the given 12-ogon, extending the V-edge that ends in vertex 10.

3 Inflate-Cut and Inflate-Paste Methods

In [12], we show that every grid n-ogon may be generated from the unit square (i.e., from the grid 4-ogon) using r Inflate-Cut transformations. We may now show exactly the same result for Inflate-Paste. At iteration k, both methods construct a grid $(2k + 4)$-ogon from the grid $(2(k - 1) + 4)$-ogon obtained in the previous iteration, for $1 \leq k \leq r$. The Inflate-Cut method yields a random grid n-ogon, if cells and rectangles are chosen at random. This is also true for Inflate-Paste, though now for the selections of v_i and of C in FSN(v_i). These algorithms are described in more detail in Sect. 4.

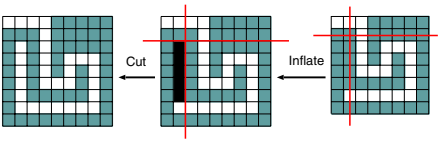


Fig. 5. The rightmost polygon is the unique grid 16-ogon that gives rise to this 18-ogon, if we apply INFLATE-CUT.

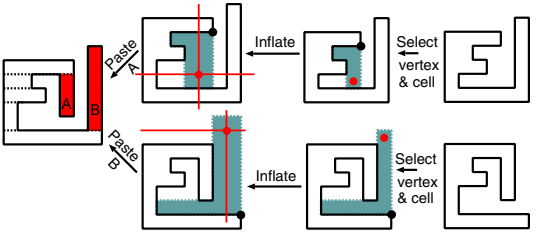


Fig. 6. The two rightmost grid 14-ogons are the unique ones that yield the 16-ogon on the left, by INFLATE-PASTE. It is also depicted $\mathrm{FSN}(v_i)$ for the two cases.

3.1 Correctness and Completeness

It is not difficult to see that both INFLATE-CUT and INFLATE-PASTE yield grid ogons. In contrast, the proof of their completeness is not immediate, as suggested by the examples given in Figs. 5 and 6.

For the proof, we need to introduce some definitions and results. Given a simple orthogonal polygon P without holes, $\Pi_{\mathrm{H}}(P)$ represents the H-decomposition of P into rectangles obtained by extending all H-edges incident to reflex vertices towards the interior of P until they hit its boundary. Each *chord* (i.e., edge extension) separates exactly two adjacent pieces (faces), since it makes an *H-cut* (see e.g. [7]). The dual graph of $\Pi_{\mathrm{H}}(P)$ captures the adjacency relation between pieces of $\Pi_{\mathrm{H}}(P)$. Its nodes are the pieces of $\Pi_{\mathrm{H}}(P)$ and its non-oriented edges connect adjacent pieces. Surely, the V-decomposition $\Pi_{\mathrm{V}}(P)$ has identical properties.

Lemma 1. *The dual graph of $\Pi_{\mathrm{H}}(P)$ is a tree for all simple orthogonal polygons P without holes.*

Proof. This result follows from the well-known Jordan Curve theorem. Suppose the graph contains a simple cycle $F_0, F_1, \ldots, F_d, F_0$, with $d \geq 2$. Let $\gamma = (\gamma_{0,1}\gamma_{1,2}\ldots\gamma_{d,0})$ be a simple closed curve in the interior of P that links the centroids of the faces $F_0, F_1, \ldots, F_d$. Denote by v the reflex vertex that defines the chord $\overline{v\,s_v}$, which separates F_0 from F_1. Here, s_v is the point where this edge's extension intersects the boundary of P. Either v or s_v would be in the interior of γ, because γ needs to cross the H-line supporting $\overline{v\,s_v}$ at least twice and only $\gamma_{0,1}$ crosses $\overline{v\,s_v}$. But the interior of γ is contained in the interior of P, and there exist points in the exterior of P which are in the neighbourhood of v and of s_v, and so we achieve a contradiction. □

We may now prove that INFLATE-PASTE is complete.

Proposition 1. *For each grid $(n+2)$-ogon, with $n \geq 4$, there is a grid n-ogon that yields it by* INFLATE-PASTE.

Proof. Given a grid $(n+2)$-ogon P, we use Lemma 1 to conclude that the dual graph of $\Pi_{\mathrm{H}}(P)$ is a tree. Each leaf of this tree corresponds to a rectangle that could have been glued by PASTE to yield P. Indeed, suppose that $\overline{v\, s_v}$ is the chord that separates a leaf F from the rest of P. Because grid ogons are in general position, s_v is not a vertex of P. It belongs to the relative interior of an edge of P. The vertex of F that is not adjacent to s_v would be $\tilde{c}$ in INFLATE-PASTE. If we cut F, we would obtain an inflated n-ogon, that we may deflate to get a grid n-ogon that yields P. The two grid lines $y = y_{\tilde{c}}$ and $x = x_{\tilde{c}}$ are free. Clearly s_v is the vertex we called v_i in the description of INFLATE-PASTE (more accurately, s_v is $\tilde{v}_i$) and $(p, q) \equiv (x_{\tilde{c}} - 1, y_{\tilde{c}} - 1) \in \mathrm{FSN}(v_i)$. □

For this paper to be self-contained, we now give a proof of the completeness of INFLATE-CUT, already sketched in [12]. It was inspired by work on convexification of simple polygons [3,10,13], in particular, by a recent paper of O. Aichholzer et al. [1]. It also shares ideas of a proof of Meisters' *Two-Ears theorem* [6] by O'Rourke. Fig. 7 shows the main ideas. A *pocket* of a nonconvex polygon P is a maximal sequence of edges of P disjoint from its convex hull except at the endpoints. The line segment joining the endpoints of a pocket is its *lid*. Any nonconvex polygon P has at least one pocket. Each pocket of an n-ogon, together with its lid, defines a simple polygon without holes, that is almost orthogonal except for an edge (lid). It is possible to slightly transform it to obtain an orthogonal polygon, as illustrated in Fig. 7. We shall refer to this polygon as an *orthogonalized pocket*.

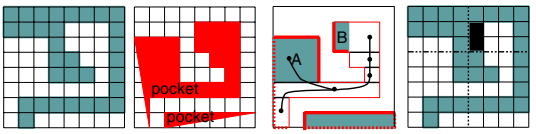
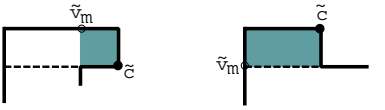


Fig. 7. The two leftmost grids show a grid 18-ogon and its pockets. The shaded rectangles A and B are either leaves or contained in leaves of the tree associated to the H-partitioning of the largest pocket. The rightmost polygon is an inflated grid 16-ogon that yields the represented grid 18-ogon, if CUT removes rectangle B.

Proposition 2. *For each grid $(n+2)$-ogon, there is a grid n-ogon that yields it by* INFLATE-CUT.

Proof. Given a grid $(n+2)$-ogon P, let Q be an orthogonalized pocket of P. Necessarily, Q is in general position. By Lemma 1 the dual graph of $\Pi_{\mathrm{H}}(Q)$ is a tree. We claim that at least one of its leaves contains or is itself a rectangle that might have been removed by CUT to yield P. Indeed, the leaves are of the two following forms, the shaded rectangles being the ones that might have been cut.



We have also represented the points that would be $\tilde{v}_m$ and $\tilde{c}$ in INFLATE-CUT. Here, we must be careful about the leaves that the lid intersects, to be able to conclude that $\tilde{c}$ is a vertex of P and that P resulted from an n-ogon in general position. Actually, an artificial H-edge, say h_Q, was introduced to render Q orthogonal, as well as an artificial V-edge. Each leaf that does not contain h_Q contains (or is itself) a rectangle that might have been removed by CUT. Every non-degenerated tree has at least two leaves. At most one leaf contains h_Q. Moreover, if the tree is degenerated (c.f. the smallest pocket in Fig. 7), then it is a leaf that could be filled. □

The notion of *mouth* [13] was crucial to reach the current formulation of CUT. Actually, INFLATE-CUT is somehow doing the reverse of an algorithm given by Toussaint that finds the convex hull of a polygon globbing-up mouths to successively remove its concavities [13]. For orthogonal polygons, we would rather define rectangular mouths. A reflex vertex v_i of an ogon P is *a rectangular mouth* of P iff the interior of the rectangle defined by v_{i-1} and v_{i+1} is in the exterior of P and neither this rectangle nor its interior contains vertices of P, except v_{i-1}, v_i and v_{i+1}. When we apply CUT to obtain a grid $(n+2)$-ogon, the vertex $\tilde{c}$ (that was the center the inflated grid cell C) is always a rectangular mouth of the resulting $(n+2)$-ogon. Thus, the proof of Proposition 2 presented above justifies Corollary 1, which rephrases the *One-Mouth theorem* by Toussaint.

Corollary 1. *Each grid n-ogon has at least one rectangular mouth, for $n \geq 6$.*

4 Quadratic-Time and Linear-Space Complexity

Our pseudocode for the two functions that yield a random grid n-ogon using INFLATE-CUT or INFLATE-PASTE is as follows, where REPLACE$(\tilde{v}, \gamma, \tilde{P})$ means replace $\tilde{v}$ by chain γ in $\tilde{P}$.

```
RANDOM-INFLATE-CUT(n)
  r := n/2 − 2
  P := {(1,1),(1,2),(2,2),(2,1)}  /* (the unit square) */
  while r > 0 do
    repeat
      Select one cell C in the interior of P (at random)
      c := the center of C
      S := {points of P first shot by H-rays and V-rays emanating from c}
      A := {v_m | vertex v_m of P satisfies the CUT-condition for C}
    until A ≠ { }
    (p, q) := the northwest corner of C
    Select v_m from A (at random)  /* v_m is (x_m, y_m) */
```

```
    e_H(v_m) := the H-edge of P that contains v_m
    Apply INFLATE using (p, q) to obtain P̃
    if e_H(v_m) = v_{m-1}v_m then
      P := REPLACE(ṽ_m, [(p+1, ỹ_m), (p+1, q+1), (x̃_m, q+1)], P̃)
    else P := REPLACE(ṽ_m, [(x̃_m, q+1), (p+1, q+1), (p+1, ỹ_m)], P̃)
    r := r − 1
  return P

RANDOM-INFLATE-PASTE(n)
  r := n/2 − 2
  P := {(1,1), (1,2), (2,2), (2,1)}
  A := P  /* (convex vertices) */
  while r > 0 do
    Select v_i from A (at random)  /* v_i is (x_i, y_i) */
    e_H(v_i) := the H-edge of P that contains v_i
    Compute FSN(v_i)
    Select cell C from FSN(v_i) (at random)
    (p, q) := the northwest corner of C
    Apply INFLATE using (p, q) to obtain P̃, Ã and ẽ_H(v_i)
    if e_H(v_i) = v_i v_{i+1} then
      P := REPLACE(ṽ_i, [(x̃_i, q+1), (p+1, q+1), (p+1, ỹ_i)], P̃)
    else P := REPLACE(ṽ_i, [(p+1, ỹ_i), (p+1, q+1), (x̃_i, q+1)], P̃)
    A := (Ã \ {ṽ_i}) ∪ {(x̃_i, q+1), (p+1, q+1)}
    if (p+1, ỹ_i) is not inside ẽ_H(v_i) then A := A ∪ {(p+1, ỹ_i)}
    r := r − 1
  return P
```

In RANDOM-INFLATE-CUT(n), "vertex v_m of P satisfies the CUT-condition" iff v_m is an extreme point of an edge of P that contains s, for some $s \in S$, and the rectangle defined by c and v_m does not contain other vertices of P except v_m.

Our implementation of RANDOM-INFLATE-CUT(n) uses linear space in n and runs in quadratic time in average. It yields a random grid 1000-ogon in 1.6 seconds in average (AMD Athlon Processor at 900 MHz). To achieve this, it keeps the vertices of P in a circular doubly linked list and keeps the total number of grid cells in the interior P per horizontal grid line (also in a linked list), but keeps no explicit representation of the grid. In addition, it keeps the current area of P (i.e., the number of cells), so that to select cell C it chooses only a positive integer less than or equal to the area. Cells in the interior of P are enumerated by rows from top to bottom. To locate C (i.e., its northwest corner (p, q)) the program uses the counters of number of cells per row to find row q and then left-to-right and top-to-bottom sweeping techniques to find the column p and the four delimiting edges. It is important to note that the V-edges (H-edges) of P that intersect each given horizontal (vertical) line occur always in couples, as shown in Fig. 8. This feature is used by the program to improve efficiency. To check whether a rectangle may be cut, the program performs a rotational sweep of the vertices of P. After each INFLATE or CUT transformation the counters and

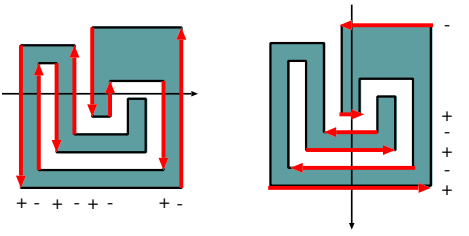


Fig. 8. Orientation of edges of P intersecting an H- or V-line.

the area of the resulting polygon are updated. INFLATE first creates a counter for the new H-line, with the same value as the counter of the previous row. Then, it analyses the sequence of H-edges that would intersect the new (imaginary) V-line, to increase counters accordingly. When a rectangle is removed, the row counters are updated by substracting the width of the rectangle removed from all counters associated with the involved rows.

Although we did not implement RANDOM-INFLATE-PASTE(n) yet, it is not difficult to see that FSN(v_i) may be found in linear time. As we mentioned in Sect. 2.3, one possibility is to follow a sweep approach, adapting an algorithm described in [9]. We assume that the H-edges and V-edges are kept sorted by y-coordinate and x-coordinate, respectively, in doubled linked lists, to simplify insertion, updating and ray shooting. To compute FSN(v_i), we determine the point u shot by the V-ray emanating from v_i to the exterior of P. This point is either on an H-edge of P or on one of the two H-lines that are free in the extended grid. Then, we move a sweep V-line from v_i to the other vertex of $e_{\mathrm{H}}(v_i)$ (possibly passing it), shrinking the *visibility interval* if some event (vertex or V-edge) obstructs visibility, until the interval becomes a singleton (i.e., $[y_i, y_i]$). The initial interval corresponds to the V-segment defined by v_i and u. Using the V-decomposition of FSN(v_i) and its area, we may select and locate C also in linear time.

5 Conclusions

We prove that every orthogonal polygon in general position may be constructed by applying either a sequence of INFLATE-CUT or INFLATE-PASTE transformations, using linear space. Each transformation may be performed in linear time using horizontal and vertical sweep, so that the construction requires quadratic-time in average for INFLATE-CUT and in the worst case for INFLATE-PASTE. These methods, in particular INFLATE-PASTE, helped us prove some interesting properties of these kind of polygons [2] and may be easily adapted to generate simple orthogonal polygons with holes. Indeed, each hole is an orthogonal polygon without holes. We are studying whether the methods may be simplified.

References

1. Aichholzer, O., Cortés, C., Demaine, E.D., Dujmovic, V., Erickson, J., Meijer, H., Overmars, M., Palop, B., Ramaswawi, S., Toussaint, G.T.: Flipturning polygons. Discrete Comput. Geom. **28** (2002) 231–253.
2. Bajuelos, A.L., Tomás, A.P., Marques, F.: Partitioning orthogonal polygons by extension of all edges incident to reflex vertices: lower and upper bounds on the number of pieces. In Proc. of ICCSA 2004. LNCS, Springer-Verlag (this volume).
3. Erdös, P.: Problem number 3763. American Mathematical Monthly **42** (1935) 627.
4. Joe, B., Simpson, R.B: Corrections to Lee's visibility polygon algorithm. BIT **27** (1987) 458–473.
5. Lee, D.T.: Visibility of a simple polygon. Computer Vision, Graphics, and Image Processing **22** (1983) 207–221.
6. Meisters, G.H.: Polygons have ears. Am. Math. Mon. **82** (1975) 648–651.
7. O'Rourke, J.: An alternate proof of the rectilinear art gallery theorem. J. Geometry **21** (1983) 118–130.
8. O'Rourke, J., Pashchenko, I., Tewari, G.: Partitioning orthogonal polygons into fat rectangles. In Proc. 13th Canadian Conference on Computational Geometry (CCCG'01) (2001) 133–136.
9. Overmars, M., Wood, D.: On rectangular visibility. J. Algorithms **9** (1988) 372–390.
10. Sz.-Nagy, B.: Solution of problem 3763. Am. Math. Mon. **46** (1939) 176–177.
11. Tomás, A.P., Bajuelos, A.L., Marques, F.: Approximation algorithms to minimum vertex cover problems on polygons and terrains. In P.M.A Sloot et al. (eds): Proc. of ICCS 2003, LNCS 2657, Springer-Verlag, (2003) 869–878.
12. Tomás, A.P., Bajuelos, A.L.: Generating Random Orthogonal Polygons. In Post-conference Proc. of CAEPIA-TTIA'2003, LNAI, Springer-Verlag (to appear).
13. Toussaint, G.T.: Polygons are anthropomorphic. Am. Math. Mon. **122** (1991) 31–35.

Partitioning Orthogonal Polygons by Extension of All Edges Incident to Reflex Vertices: Lower and Upper Bounds on the Number of Pieces*

António Leslie Bajuelos[1], Ana Paula Tomás[2], and Fábio Marques[3]

[1] Dept. of Mathematics & CEOC - Center for Research in Optimization and Control, University of Aveiro, Portugal
leslie@mat.ua.pt
[2] DCC-FC & LIACC, University of Porto, Portugal
apt@ncc.up.pt
[3] School of Technology and Management, University of Aveiro, Portugal
fabio@estga.ua.pt

Abstract. Given an orthogonal polygon P, let $|\Pi(P)|$ be the number of rectangles that result when we partition P by extending the edges incident to reflex vertices towards $\mathrm{INT}(P)$. In [4] we have shown that $|\Pi(P)| \leq 1 + r + r^2$, where r is the number of reflex vertices of P. We shall now give sharper bounds both for $\max_P |\Pi(P)|$ and $\min_P |\Pi(P)|$. Moreover, we characterize the structure of orthogonal polygons in general position for which these new bounds are exact. We also present bounds on the area of grid n-ogons and characterize those having the largest and the smallest area.

1 Introduction

We shall call a *simple polygon* P a region of a plane enclosed by a finite collection of straight line segments forming a simple cycle. Non-adjacent segments do not intersect and two adjacent segments intersect only in their common endpoint. These intersection points are the *vertices* of P and the line segments are the *edges* of P. This paper deals only with simple polygons, so that we simply call just polygons, in the sequel. We will denote the interior of the polygon P by $\mathrm{INT}(P)$ and the boundary by $\mathrm{BND}(P)$. The boundary is thus considered part of the polygon; that is, $P = \mathrm{INT}(P) \cup \mathrm{BND}(P)$. A vertex is called *convex* if the interior angle between its two incident edges is at most π; otherwise it is called *reflex* (or *concave*). We use r to represent the number of reflex vertices of P. A polygon is called *orthogonal* (or *rectilinear*) iff its edges meet at right angles. O'Rourke [3] has shown that $n = 2r + 4$ for every n-vertex orthogonal polygon (n-*ogon*, for short). So, orthogonal polygons have an even number of vertices.

* Partially funded by LIACC through *Programa de Financiamento Plurianual, Fundação para a Ciência e Tecnologia (FCT)* and *Programa POSI*, and by CEOC (Univ. of Aveiro) through *Programa POCTI, FCT, co-financed by EC fund FEDER.*

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 127–136, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Definition 1. *A rectilinear cut (r-cut) of an n-ogon P is obtained by extending each edge incident to a reflex vertex of P towards* $\mathrm{INT}(P)$ *until it hits* $\mathrm{BND}(P)$. *We denote this partition by $\Pi(P)$ and the number of its elements (pieces) by $|\Pi(P)|$. Each piece is a rectangle and so we call it a* r-piece.

In [4] we proposed an algorithm to solve the Minimum Vertex Guard problem for polygons, whose main idea is to enclose the optimal solution within intervals that are successively shortened. To find these intervals, it goes on refining a decomposition of the polygon and solving optimization problems that are smaller than the original one. To improve efficiency, it tries to take advantage of the polygon's topology and, in particular, of the fact that some pieces in the decomposition may be dominant over others (i.e., if they are visible so are the dominated ones). The finer the decomposition is, the better the approximation becomes, but the problem that the algorithm has to solve at each step might become larger. For the case of orthogonal polygons, we could start from different partitions, one of which is $\Pi(P)$ and so, we were interested in establishing more accurate bounds for the number of pieces that $\Pi(P)$ might have in general.

The paper is structured as follows. We then introduce some preliminary definitions and useful results. In Sect. 2, we present the major result of this paper, that establishes lower and upper bounds on $|\Pi(P)|$, and improves an upper bound we gave in [4]. Finally, Sect. 3 contains some interesting results about lower and upper bounds on the area of grid n-ogons, although they do not extend to generic orthogonal polygons.

1.1 Preliminaries

Generic orthogonal polygons may be obtained from a particular kind of orthogonal polygons, that we called grid orthogonal polygons, as depicted in Fig. 1. (The reader may skip Definition 2 and Lemmas 1 and 2 if he/she has read [5].)

Definition 2. *An n-ogon P is in general position iff every horizontal and vertical line contains at most one edge of P, i.e., iff P has no collinear edges. We call "grid n-ogon" to each n-ogon in general position defined in a $\frac{n}{2} \times \frac{n}{2}$ square grid.*

Lemma 1 follows immediately from this definition.

Lemma 1. *Each grid n-ogon has exactly one edge in every line of the grid.*

Each n-ogon which is not in general position may be mapped to an n-ogon in general position by ϵ-perturbations, for a sufficiently small constant $\epsilon > 0$. Consequently, we shall first address n-ogons in general position.

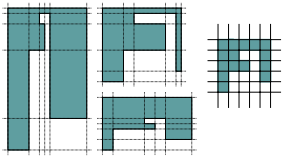


Fig. 1. Three 12-ogons mapped to the same grid 12-ogon.

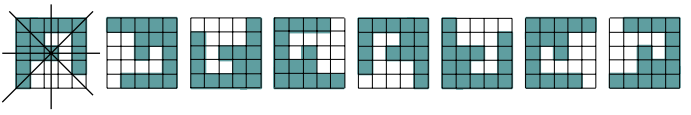


Fig. 2. Eight grid n-ogons that are symmetrically equivalent. From left to right, we see images by clockwise rotations of 90°, 180° and 270°, by flips wrt horizontal and vertical axes and flips wrt positive and negative diagonals.

Lemma 2. *Each n-ogon in general position is mapped to a unique grid n-ogon through top-to-bottom and left-to-right sweep. And, reciprocally, given a grid n-ogon we may create an n-ogon that is an instance of its class by randomly spacing the grid lines in such a way that their relative order is kept.*

The number of classes may be further reduced if we group grid n-ogons that are symmetrically equivalent. In this way, the grid n-ogons in Fig. 2 represent the same class.

Given an n-ogon P in general position, $\textsc{Free}(P)$ represents any grid n-ogon in the class that contains the grid n-ogon to which P is mapped by the sweep procedure described in Lemma 2.

The following result is a trivial consequence of the definition of $\textsc{Free}(P)$.

Lemma 3. *For all n-ogons P in general position, $|\Pi(P)| = |\Pi(\textsc{Free}(P))|$.*

2 Lower and Upper Bounds on $|\Pi(P)|$

In [4] we have shown that $\Pi(P)$ has at most $1 + r + r^2$ pieces. Later we noted that this upper bound is not sufficiently tightened. Actually, for small values of r, namely $r = 3, 4, 5, 6, 7$, we experimentally found that the difference between $1 + r + r^2$ and $\max|\Pi(P)|$ was 1, 2, 4, 6 and 9, respectively.

Definition 3. *A grid n-ogon Q is called* Fat *iff $|\Pi(Q)| \geq |\Pi(P)|$, for all grid n-ogons P. Similarly, a grid n-ogon Q is called* Thin *iff $|\Pi(Q)| \leq |\Pi(P)|$, for all grid n-ogons P.*

The experimental results supported our conjecture that there was a single Fat n-ogon (except for symmetries of the grid) and that it had the form illustrated in Fig. 3.

Clearly, each r-piece is defined by four vertices. Each vertex is either in $\text{INT}(P)$ (*internal vertex*) or is in $\text{BND}(P)$ (*boundary vertex*). Similar definitions hold for the edges. An edge e of r-piece R is called an *internal edge* if $e \cap \text{INT}(P) \neq \emptyset$, and it is called a *boundary edge* otherwise.

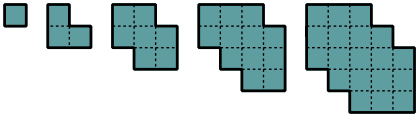


Fig. 3. The unique Fat n-ogons (symmetries excluded), for $n = 4, 6, 8, 10, 12$.

Lemma 4. *The total number $|V_i|$ of internal vertices in $\Pi(P)$, when the grid n-ogon P is as illustrated in Fig. 3 is given by (1) where r is the number of reflex vertices of P.*

$$|V_i| = \begin{cases} \frac{3r^2-2r}{4} & \text{for } r \text{ even.} \\ \frac{(3r+1)(r-1)}{4} & \text{for } r \text{ odd.} \end{cases} \quad (1)$$

Proof. By construction, $|V_i|$ is defined by (2).

$$|V_i| = \begin{cases} 2\sum_{k=1}^{\frac{r}{2}}(r-k) & \text{iff } r \text{ is even.} \\ (r-\frac{r+1}{2}) + 2\sum_{k=1}^{\frac{r-1}{2}}(r-k) & \text{iff } r \text{ is odd.} \end{cases} \quad (2)$$

□

Proposition 1. *If P is any n-vertex orthogonal polygon such that the number of internal vertices of $\Pi(P)$ is given by (1), then P has at most a single reflex vertex in each horizontal and vertical line.*

Proof. We shall suppose first that P is a grid n-ogon. Then, let $v_{L_1} = (x_{L_1}, y_{L_1})$ and $v_{R_1} = (x_{R_1}, y_{R_1})$ be one of the leftmost and one of the rightmost reflex vertices of P, respectively. The horizontal chord with origin at v_{L_1} can intersect at most $x_{R_1} - x_{L_1}$ vertical chords, since we shall not count the intersection with the vertical chord defined by v_{L_1}. The same may be said about the horizontal chord with origin at v_{R_1}. There are exactly r vertical and r horizontal chords, and thus $x_{R_1} - x_{L_1} \leq r-1$. If there were c vertical edges such that both extreme points are reflex vertices then $x_{R_1} - x_{L_1} \leq r - 1 - c$. This would imply that the number of internal vertices of $\Pi(P)$ would be strictly smaller than the value defined by (1). In fact, we could proceed to consider the second leftmost vertex (for $x > x_{L_1}$), say v_{L_2}, then second rightmost vertex (for $x < x_{R_1}$), and so forth. The horizontal chord that v_{L_2} defines either intersects only the vertical chord defined by v_{L_1} or it does not intersect it at all. So, it intersects at most $r-2-c$ vertical chords. In sum, c should be null, and by symmetry, we would conclude that there is exactly one reflex vertex in each vertical grid line (for $x > 1$ and $x < \frac{n}{2} = r + 2$).

Now, if P is not a grid n-ogon but is in general position, then $\Pi(P)$ has the same combinatorial structure as $\Pi(\text{FREE}(P))$, so that we do not have to prove anything else. If P is not in general position, then let us render it in general position by a sufficiently small ϵ-perturbation, so that the partition of this latter polygon does not have less internal vertices than $\Pi(P)$. □

Corollary 1. *For all grid n-ogons P, the number of internal vertices of $\Pi(P)$ is less than or equal to the value established by (1).*

Proof. It results from the proof of Proposition 1. □

Theorem 1. *Let P be a grid n-ogon and $r = \frac{n-4}{2}$ the number of its reflex vertices. If P is* Fat *then*

$$|\Pi(P)| = \begin{cases} \frac{3r^2+6r+4}{4} & \text{for } r \text{ even} \\ \frac{3(r+1)^2}{4} & \text{for } r \text{ odd} \end{cases}$$

and if P is Thin *then $|\Pi(P)| = 2r + 1$.*

Proof. Suppose that P is a grid n-ogon. Let V, E and F be the sets of all vertices, edges and faces of $\Pi(P)$, respectively. Let us denote by V_i and V_b the sets of all internal and boundary vertices of the pieces of $\Pi(P)$. Similarly, E_i and E_b represent the sets of all internal and boundary edges of such pieces. Then, $V = V_i \cup V_b$ and $E = E_i \cup E_b$. Since P is in general position, each chord we draw to form $\Pi(P)$ hits $\mathrm{BND}(P)$ in the interior of an edge and no two chords hit $\mathrm{BND}(P)$ in the same point. Hence, using O'Rourke's formula [3] we obtain $|E_b| = |V_b| = (2r + 4) + 2r = 4r + 4$. We may easily see that to obtain Fat n-ogons we must maximize the number of internal vertices.

By Corollary 1,

$$\max_P |V_i| = \begin{cases} \frac{3r^2-2r}{4} & \text{for } r \text{ even} \\ \frac{(3r+1)(r-1)}{4} & \text{for } r \text{ odd} \end{cases}$$

and, therefore, $\max_P |V| = \max_P(|V_i| + |V_b|)$ is given by

$$\max_P |V| = \begin{cases} \frac{3r^2+14r+16}{4} & \text{for } r \text{ even} \\ \frac{3r^2+14r+15}{4} & \text{for } r \text{ odd} \end{cases}$$

From Graph Theory [1] we know that the sum of the degrees of the vertices in a graph is twice the number of its edges, that is, $\sum_{v \in V} \delta(v) = 2|E|$. Using the definitions of grid n-ogon and of $\Pi(P)$, we may partition V as

$$V = V_c \cup V_r \cup (V_b \setminus (V_c \cup V_r)) \cup V_i$$

where V_r and V_c represent the sets of reflex and of convex vertices of P, respectively. Moreover, we may conclude that $\delta(v) = 4$ for all $v \in V_r \cup V_i$, $\delta(v) = 3$ for all $v \in V_b \setminus (V_c \cup V_r)$ and $\delta(v) = 2$ for all $v \in V_c$. Hence,

$$\begin{aligned} 2|E| &= \textstyle\sum_{v \in V_r \cup V_i} \delta(v) + \sum_{v \in V_c} \delta(v) + \sum_{v \in V_b \setminus (V_c \cup V_r)} \delta(v) \\ &= 4|V_i| + 4|V_r| + 2|V_c| + 3(|V_b| - |V_r| - |V_c|) = 4|V_i| + 12r + 8 \end{aligned}$$

and, consequently, $|E| = 2|V_i| + 6r + 4$.

Similarly, to obtain Thin n-ogons we must minimize the number of internal vertices of the arrangement. There are grid n-ogons such that $|V_i| = 0$, for all n (such that $n = 2r + 4$ for some $r \geq 0$). Thus, for Thin n-ogons $|V| = 4r + 4$.

Finally, to conclude the proof, we have to deduce the expression of the upper and lower bound of the number of faces of $\Pi(P)$, that is of $|\Pi(P)|$. Using Euler's formula $|F| = 1 + |E| - |V|$, and the expressions deduced above, we have $\max_P |F| = 1 + 2(\max_P |V_i|) + 6r + 4 - \max_P |V|$. That is, $\max_P |F| = \max_P |V_i| + 6r + 5$, so that

$$\max_P |F| = \begin{cases} \frac{3r^2+6r+4}{4} & \text{for } r \text{ even} \\ \\ \frac{3(r+1)^2}{4} & \text{for } r \text{ odd} \end{cases}$$

and $\min_P |F| = 1+2(\min_P |V_i|)+6r+4-\min_P |V| = 1+6r+4-4r-4 = 2r+1$. The existence of FAT and THIN grid n-ogons, for all n, follows from Lemma 4 and from the construction indicated in Fig. 6, respectively. □

Fig. 4 shows some THIN n-ogons.

Fig. 4. Some grid n-gons with $|V_i| = 0$.

The area of a grid n-ogon is the number of grid cells in its interior. Corollary 2 gives some more insight into the structure of FATs, although the stated condition is not sufficient for a grid ogon to be FAT.

Corollary 2. *If P is a* FAT *grid n-ogon then each r-piece in $\Pi(P)$ has area 1.*

Proof. By Pick's Theorem (see, e.g. [2]), the area $A(P)$ of grid n-ogon P is given by (3)

$$A(P) = \frac{b(P)}{2} + i(P) - 1 \tag{3}$$

where $b(P)$ and $i(P)$ represent the number of grid points contained in BND(P) and INT(P), respectively. Using (3) and the expressions deduced in Theorem 1, we conclude that if P is FAT then

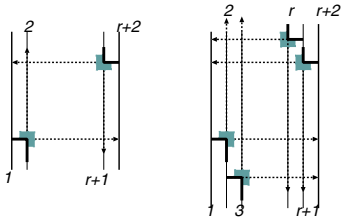
$$A(P) = \begin{cases} \frac{4r+4}{2} + \frac{3r^2-2r}{4} - 1 = \frac{3r^2+6r+4}{4} & \text{for } r \text{ even} \\ \\ 2r+1+\frac{(3r+1)(r-1)}{4} = \frac{3(r+1)^2}{4} & \text{for } r \text{ odd} \end{cases}$$

so that $A(P) = |\Pi(P)|$. Hence, each r-piece has area 1. □

Nevertheless, based on the proof of Proposition 1, we may prove the uniqueness of FATs and fully characterize them.

Proposition 2. *There is a single* FAT *n-ogon (except for symmetries of the grid) and its form is illustrated in Fig. 3.*

Proof. We saw that FAT n-ogons must have a single reflex vertex in each vertical grid-line, for $x > 1$ and $x < \frac{n}{2}$. Also, the horizontal chords with origins at the reflex vertices that have $x = 2$ and $x = \frac{n}{2} - 1 = r + 1$, determine $2(r-1)$ internal points (by intersections with vertical chords). To achieve this value, they must be positioned as illustrated below on the left.



Moreover, the reflex vertices on the vertical grid-lines $x = 3$ and $x = r$ add $2(r-2)$ internal points. To achieve that, we may conclude by some simple case reasoning, that v_{L_2} must be below v_{L_1} and v_{R_2} must be above v_{R_1}, as shown above on the right. And, so forth... □

FAT n-grid ogons are not the grid n-ogons that have the largest area, except for small values of n, as we may see in Fig 5. Some more details are given in the following section, where we shall also prove that the set of grid ogons that have the smallest area is a proper subset of the THIN grid ogons.

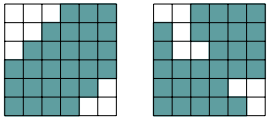


Fig. 5. On the left we see the FAT grid 14-ogon. It has area 27, whereas the grid 14-ogon on the right has area 28, which is the maximum for $n = 14$.

3 Lower and Upper Bounds on the Area

In [5] we proposed an iterative method that constructs a grid n-ogon from the unit square by applying a transformation we called INFLATE-PASTE r times. Based on this method we may show the following result.

Lemma 5. *Each $(i+2)$-grid ogon P is obtained by adding at least two grid cells to a i-grid ogon (dependent of P), for all even $i \geq 4$.*

Proof. Each INFLATE-PASTE transformation increases the area of the grid ogon constructed up to a given iteration by at least two (i.e., it glues at least two grid cells to the polygon) and the INFLATE-PASTE method is complete. □

Another concept is needed for our proof of Proposition 3, stated below. A *pocket* of a nonconvex polygon P is a maximal sequence of edges of P disjoint

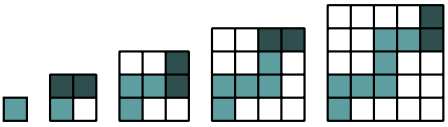


Fig. 6. Constructing the grid ogons of the smallest area, for $r = 0, 1, 2, 3, 4, \ldots$. The area is $2r + 1$.

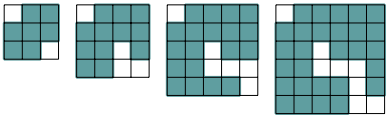


Fig. 7. A family of grid n-ogons with MAX-AREA (the first elements: $r = 2, 3, 4, 5, \ldots$).

from its convex hull except at the endpoints. The line segment joining the endpoints of a pocket is called its *lid*. Any nonconvex polygon P has at least one pocket. Each pocket of an n-ogon, together with its lid, defines a simple polygon without holes, that is almost orthogonal except for an edge (lid). It is possible to slightly transform it to obtain an orthogonal polygon, say an *orthogonalized pocket*.

We may now prove the following property about the area of grid ogons. We note that, for $r = 1$, there is a single grid ogon (except for symmetries of the grid) which is necessarily the one with the smallest and the largest area.

Proposition 3. *Let P_r be a grid n-ogon and $r = \frac{n-4}{2}$ the number of its reflex vertices. Then $2r + 1 \leq A(P_r) \leq r^2 + 3$, for $r \geq 2$.*

Proof. From Lemma 5, we may conclude that $A(P_r) \geq 2r + 1$, for all P_r and all $r \geq 1$. The INFLATE-PASTE method starts from the unit square (that is P_0) and applies r INFLATE-PASTE transformations to construct P_r. In each transformation it glues two cells (at least) to the polygon being constructed, so that $A(P_r) \geq 2r + 1$. Fig. 6 may provide some useful intuition to this proof. To prove that $A(P_r) \leq r^2 + 3$, we imagine that we start from a square (not grid ogon) of area $(r + 1)^2$. This is equivalent to saying that our $\frac{n}{2} \times \frac{n}{2}$ square grid (that consists of $(r+1)^2$ unit square cells) is initially completely filled. Then, we start removing grid cells, to create reflex vertices, while keeping the orthogonal polygon in the grid in general position. Each time we introduce a new reflex vertex, we are either creating a new pocket or increasing a pocket previously created. To keep the polygon in general position, only two pockets may start at the corners (indeed opposite corners) and to start each one we delete one cell (at least). To create any other pocket we need to delete at least three cells. On the other hand, by Lemma 5, to augment an already created pocket, we have to delete at least two cells. In sum, to obtain a polygon with the maximal area we have to remove the smallest number of cells, so that only two pockets may be created. Each one must be a grid ogon with the smallest possible area. In Fig. 7 we show a family of polygons that have the largest area, $A(P_r) = r^2 + 3$. □

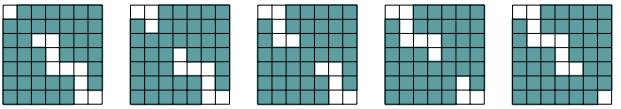


Fig. 8. A sequence of MAX-AREA n-ogons, for $n = 16$.

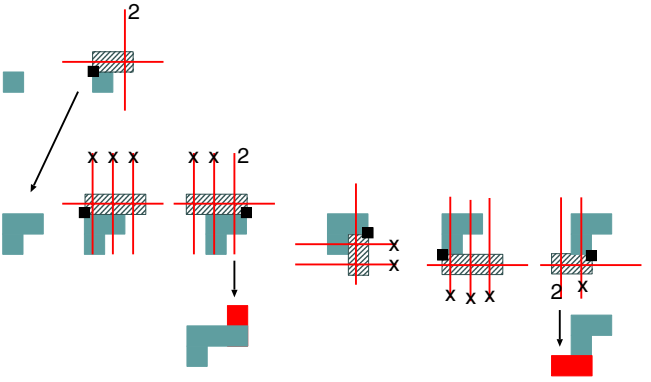


Fig. 9. Uniqueness of MIN-AREA grid n-ogons related to INFLATE-PASTE construction.

Definition 4. *A grid n-ogon P is a* MAX-AREA *grid n-ogon iff $A(P) = r^2 + 3$ and it is a* MIN-AREA *grid n-ogon iff $A(P) = 2r + 1$.*

There exist MAX-AREA grid n-ogons for all n, as indicated in Fig. 7, but they are not unique, as we may see in Fig. 8. Regarding MIN-AREA n-ogons, it is obvious that they are THIN grid n-ogons, because $|\Pi(P)| = 2r + 1$ holds only for THIN grid n-ogons. This condition is not sufficient for a grid n-ogon to be a MIN-AREA grid n-ogon (see for example the rightmost grid n-ogon in Fig. 4). Based on Proposition 3 and on the INFLATE-PASTE method, we may prove the uniqueness of MIN-AREA grid n-ogons.

Proposition 4. *There is a single* MIN-AREA *grid n-ogon (except for symmetries of the grid) and it has the form illustrated in Fig. 6.*

Proof (Sketch). It is strongly based on the INFLATE-PASTE construction. The idea is to proceed by induction on r and by case analysis to see which are the convex vertices v_i that allow to increase the area by just two units (see Fig. 9). □

4 Further Work

We are now investigating how the ideas of this work may be further exploited to obtain better approximate solutions to the MINIMUM VERTEX GUARD problem, where the goal is to find the minimum number of vertex guards that are necessary to completely guard a given polygon. Our strategy is to establish bounds for families of grid ogons and to see how these bounds apply to the orthogonal polygons in the class of a given n-ogon.

References

1. Bondy, J., Murty, U.: Graph Theory with Applications. Elseiver Science, New York, (1976).
2. Fekete, S. P.: On simple polygonalizations with optimal area. Discrete & Computational Geometry **23** (2000) 73–110.
3. O'Rourke, J.: An alternate proof of the rectilinear art gallery theorem. J. of Geometry **21** (1983) 118–130.
4. Tomás, A. P., Bajuelos, A. L., Marques, F.: Approximation algorithms to minimum vertex cover problems on polygons and terrains. In P.M.A Sloot et al. (Eds): Proc. of ICCS 2003, LNCS 2657, Springer-Verlag (2003) 869–878.
5. Tomás, A. P., Bajuelos, A. L.: Quadratic-Time Linear-Space Algorithms for Generating Orthogonal Polygons with a Given Number of Vertices. In Proc. of ICCSA 2004. LNCS, Springer-Verlag (this volume).

On the Time Complexity of Rectangular Covering Problems in the Discrete Plane

Stefan Porschen

Institut für Informatik, Universität zu Köln, D-50969 Köln, Germany.
porschen@informatik.uni-koeln.de

Abstract. This paper addresses the computational complexity of optimization problems dealing with the covering of points in the discrete plane by rectangles. Particularly we prove the NP-hardness of such a problem(class) defined by the following objective function: Simultaneously minimize the total area, the total circumference and the number of rectangles used for covering (where the length of every rectangle side is required to lie in a given interval). By using a tiling argument we also prove that a variant of this problem, fixing only the minimal side length of rectangles, is NP-hard. Such problems may appear at the core of applications like data compression, image processing or numerically solving partial differential equations by multigrid computations.

Keywords: NP-completeness; rectangular set cover; discrete plane; integer lattice

1 Introduction and Motivation

There are several problem(classe)s concerning the covering of point sets by geometrical objects in the euclidean plane [3]. Such problems are geometrical variants of pure set - or graph theoretical covering problems (cf. e.g. [1,11]). Most of these geometrical covering problems (as far as dealing with arbitrary many covering components) are NP-hard [7,6,9,15] just as their set theoretical counterparts. Closely related to such covering problems are partition - and clustering problems [4,5,8]. The partition variants do not allow input points to be covered by more than one covering patch. Another class of discrete plane problems consists of tiling or packing problems [10,14]. Here, a region of the plane (e.g. the convex hull of a point set) has to be exactly packed by (non-overlapping) geometrical objects of a given shape.

In this paper we pose some variants of geometrical covering problems and study their computational complexities. Namely, we are interested in problems where a covering of planar points by rectangles is searched such that certain objective functions are minimized. More precisely, the focus lies on problems of the following kind: Given a finite set M of points (each having integer coordinates) in the euclidean plane and a positive (real) number k. Find a set R of rectangles (each having sides of length at least k) covering all points of M such that simultaneously the total area, the total circumference and the number of

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 137–146, 2004.
© Springer-Verlag Berlin Heidelberg 2004

rectangles in R is minimized. A variant of this problem is given by introducing a second parameter k' fixing also the maximal side length of rectangles.

The motivation for studying such problems stems from the numerical analysis of partial differential equations (PDE's). The PDE's modelling a given application are discretized on an integer grid (in an euclidean space of appropriate finite dimension) and are treated by so-called adaptive multigrid procedures (cf. e.g. [2]). According to the values of error estimation functions it has to be decided iteratively whether the grid has to be refined in certain regions. These refinement steps can be carried out by covering the indicated regions by regular rectangles optimized due to some reasonable cost constraints. Such computations usually are implemented on parallel machines, where for achieving reasonable running times a load balancing between the processors has to be managed thereby minimizing communication overheads. The objective function described above (for the 2-dimensional case) takes into account also the circumference of rectangles which can represent the communication costs between processor groups assigned to the rectangular regions. Other applicational fields for such problems are image processing [13,15] and data compression. On the other hand, problems as stated above are also worth to be studied abstractly which basically is the point of view in this paper.

2 Basic Definitions and Notation

Let $\mathbb{E}^2$ denote the euclidean plane which is the real vector space $\mathbb{R}^2$ equipped with the (orthogonal) standard basis $\boldsymbol{e}_x, \boldsymbol{e}_y \in \mathbb{R}^2$, and the standard scalar product inducing the norm topology. As it turns out, all problems studied in the following (having finite instances) are invariant under operations by the group of translations. Hence, we can shift a given problem instance into the first quadrant $\mathbb{E}^2_+$ of the plane by an appropriate translation, solve the problem for the shifted instance, and shift the solution configuration back by the inverse translation. Thus we assume throughout w.l.o.g. that the geometrical objects of problem instances are located in the first quadrant. (Equivalently, first fix the location of geometrical objects in the affine space, then introduce a coordinate system, so that the objects appear in the first quadrant.)

Let an *isothetical*, i.e. axis parallel, integer lattice (grid) $L_\lambda = \mathbb{Z}\boldsymbol{e}_x\lambda + \mathbb{Z}\boldsymbol{e}_y\lambda$ be embedded in $\mathbb{E}^2$, for fixed real lattice constant $\lambda > 0$. (It is convenient to keep λ as a problem parameter since applications like solving PDE's on a grid require a varying lattice constant allowing to refine grid regions (in different ways) depending on previous computational results.) Given a lattice point $z \in L_\lambda$, we refer to its coordinates by $x(z)$ respectively $y(z)$. Since we are interested only in finite problem instances we can restrict the considerations to a bounded region $B := [0, N_x\lambda] \times [0, N_y\lambda]$ of the first quadrant, for $N_x, N_y \in \mathbb{N}$. Let $I_\lambda := B \cap L_\lambda$ denote the corresponding part of the lattice.

Throughout we require that rectangles used for covering are placed isothetically in the plane. By $\ell_x(r)$ respectively $\ell_y(r)$) the length of the x-parallel respectively y-parallel side of a rectangle r is denoted. We allow only *proper* rectangles

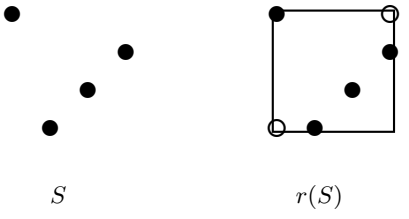


Fig. 1. Black dots represent points of S (left), white dots represent the diagonal points $z_u(S), z_d(S)$ of the rectangle $r(S)$ enclosing S (right); grid lines are omitted.

r meaning $\ell_x(r) > 0$ and $\ell_y(r) > 0$. Let $a(r)$ be the area and $u(r)$ be the circumference of r. We have to distinguish between abstract geometrical objects like rectangles or squares and concrete instances of these objects located in the plane. A rectangle $r = [x_d, x_u] \times [y_d, y_u]$ in the plane is uniquely determined by its upper right $z_u := (x_u, y_u) \in \mathbb{E}^2$ and lower left $z_d := (x_d, y_d) \in \mathbb{E}^2$ diagonal points, which must not necessarily be lattice points.

There are two possibilities for what it means that a set of points is covered by a rectangle: 1.) Points have to lie in the strict inner region of a rectangle (i.e., rectangles are regarded as open sets according to the norm topology in $\mathbb{E}^2$). 2.) Points are allowed also to lie on the boundary of rectangles (i.e., rectangles are considered to be topologically closed). In the sequel we take the latter point of view: points located on the boundary of a rectangle are defined to be covered by it. In accordance with this convention and for wasting no resources it would be reasonable to require that whenever possible a rectangle shall enclose tightly the set S of lattice points covered by it (cf. Fig. 1). Such an enclosing rectangle $r(S)$ is determined by S since its lower left and upper right diagonal points are given by $z_d(S) := (x_d(S), y_d(S))$ and $z_u(S) := (x_u(S), y_u(S))$, where $x_d(S) := \min_{z \in S} x(z)$, $y_d(S) := \min_{z \in S} y(z)$ and $x_u(S) := \max_{z \in S} x(z)$, $y_u(S) := \max_{z \in S} y(z)$. Hence $r(S)$ has edges coinciding with lattice points. Obviously this construction violates the properness condition in the case that the points of S lie all on the same grid line, since then $r(S)$ corresponds to a line segment.

Particularly, we are interested in the computational complexity of rectangular covering decision problems and also in certain optimization counterparts of them. To this end let $\mathcal{R}$ denote the *universe of rectangles* which could be used for an arbitrary covering problem in the plane. (Here the concrete objects located in the plane are meant as defined above. From a slightly different point of view $\mathcal{R}$ can also be regarded as the collection of all rectangular *types*, each defined by a pair $(\ell_x, \ell_y) \in \mathbb{R}_+^2$ fixing only the length of rectangle sides parallel to the x- resp. y-axis.) Further conditions posed by a concrete problem may restrict $\mathcal{R}$ to an appropriate subset. The heart of an optimization problem consists of its objective function. It is convenient to introduce such an objective as a map assigning costs to rectangles. To be flexible enough, we take a rather general point of view not fixing a specific class of cost functions.

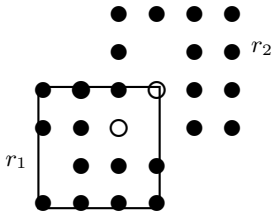


Fig. 2. Two rectangles r_1, r_2 whose intersection contains points (white dots) of the input set M (black dots); grid lines are omitted.

Definition 1. *An* objective function on rectangles *is a partial map* $w : \mathcal{R} \to \mathbb{R}_+$ *(the domain* $D(w)$ *will be made explicit by concrete problems), whose values* $w(r)$ *are assumed to be computable in constant time. Given* w*, a rectangle* r *is called* admissible *if* $r \in D(w)$*. To an objective function* w *assign the following* $\mathbb{R}_+$*-valued extension to sets defined by* $w'(R) := \sum_{r \in R} w(r)$*, for every* $R \subset D(w)$*. (Since the meaning should become clear from the context we also symbolize the set extension by* w*.) An objective function on rectangles is called* monotone *if it satisfies:*

$$\forall r, r' \in D(w) : r \subseteq r' \Rightarrow w(r) \leq w(r')$$

The monotonicity condition simply reflects the reasonable requirement that the costs contributed by a rectangle should not be decreased by a smaller rectangle.

Remark 1. *Notice that the monotonicity of* w *in general does not imply the monotonicity of its extension to sets in the sense that* $w(R) \leq w(R')$ *holds if* $R \subseteq R'$*, where* $R, R' \subset D(w)$*. Obviously, in the particular case of the constant objective function* $w \equiv c$ *(*$c \in \mathbb{R}_+$*) we also have monotonicity of the set extension.*

Every optimization problem studied in the next section searches for a certain subset $R \subset \mathcal{R}$ serving as a covering of a finite input set $M \subset I_\lambda$ of lattice points meaning $M \subseteq \bigcup_{r \in R} r \cap I_\lambda$. The rectangles of such a covering are permitted to overlap in any way, in contrast to the rules for tiling problems, where the geometrical objects should achieve an exact packing of a given region and therefore may overlap only in parts of their boundaries. It can also happen that $(r \cap r') \cap M \neq \emptyset$, i.e., there are points in M which are multiply covered, namely by r and r' (cf. Fig. 2). Such situations distinguish covering problems from rectangular partition problems allowing overlapping rectangles only in case of empty intersection with the input set M.

3 Rectangular Covering Problems and Their Computational Complexities

Next, we analyse several rectangular covering problems, which differ with regard to their input parameters and their objective functions. As a proof basis for what

follows serves a quadratic covering problem which uses squares of a prescribed fixed type:

Definition 2. *The* (fixed type) quadratic covering problem $\mathrm{QC}^{\lambda}_{\mathrm{fix}}$ *is the following search problem: For a fixed real lattice constant* $\lambda > 0$*, let a point set* $M = \{z_1, \ldots, z_n\} \subset I_\lambda$ *(*$n \in \mathbb{N}$*) and* $t \in \mathbb{R}_+, t > 0$*, be given. Find a set* Q *of (isothetical) squares of side length* t *in such that* $M \subset \bigcup_{q \in Q} q \cap I_\lambda$ *and* $|Q|$ *is minimized. In the* decision version $\mathrm{DQC}^{\lambda}_{\mathrm{fix}}$ *a further input parameter* $N \in \mathbb{N}$ *is given, and it has to be decided whether there exists a covering* Q *of* M *such that* $|Q| \leq N$*.*

Notice that $\mathrm{DQC}^{\lambda}_{\mathrm{fix}} \in$ NP. Indeed, let (M, t, N) be an arbitrary instance of $\mathrm{DQC}^{\lambda}_{\mathrm{fix}}$ and let Q be a (feasible) set of isothetical squares (delivered by an oracle). It can be easily checked in time $O(|M||Q|)$ whether $|Q| \leq N$ and whether $M \subset \bigcup_{q \in Q} q \cap I_\lambda$. Due to [9,15] we have the following assertion for the special case that the underlying lattice constant equals 1:

Proposition 1. $\mathrm{DQC}^{1}_{\mathrm{fix}}$ *is* NP*-complete and* $\mathrm{QC}^{1}_{\mathrm{fix}}$ *is* NP*-hard.* □

An obvious consequence of this result is:

Corollary 1. $\mathrm{DQC}^{\lambda}_{\mathrm{fix}}$ *is* NP*-complete and* $\mathrm{QC}^{\lambda}_{\mathrm{fix}}$ *is* NP*-hard.* □

An immediate generalization is the following fixed type rectangular covering problem prescribing both side length for rectangles.

Definition 3. *The* (fixed type) rectangular covering problem $\mathrm{RC}^{\lambda}_{\mathrm{fix}}$ *is the following search problem: For a fixed real lattice constant* $\lambda > 0$*, let a point set* $M = \{z_1, \ldots, z_n\} \subset I_\lambda$ *(*$n \in \mathbb{N}$*) and* $t, t' \in \mathbb{R}_+, t' \geq t > 0$ *be given. Find a set* R *of isothetical rectangles each having two parallel sides of length* t *resp.* t' *such that* $M \subset \bigcup_{r \in R} r \cap I_\lambda$ *and* $|R|$ *is minimized. In the* decision version $\mathrm{DRC}^{\lambda}_{\mathrm{fix}}$ *a further input parameter* $N \in \mathbb{N}$ *is given, and it has to be decided whether there exists a covering* R *of* M *such that* $|R| \leq N$*.*

Lemma 1. $\mathrm{DRC}^{\lambda}_{\mathrm{fix}}$ *is* NP*-complete and* $\mathrm{RC}^{\lambda}_{\mathrm{fix}}$ *is* NP*-hard.*

Proof. $\mathrm{DRC}^{\lambda}_{\mathrm{fix}}$ is obviously in NP. It is also NP-complete since $\mathrm{DQC}^{\lambda}_{\mathrm{fix}}$ is a special case of it: To each instance (M, t, N) of $\mathrm{DQC}^{\lambda}_{\mathrm{fix}}$, the instance $(M, t, t' = t, N)$ of $\mathrm{DRC}^{\lambda}_{\mathrm{fix}}$ can be assigned (in polynomial time). From the NP-hardness of $\mathrm{QC}^{\lambda}_{\mathrm{fix}}$ being a special case of $\mathrm{RC}^{\lambda}_{\mathrm{fix}}$ immediately the NP-hardness of $\mathrm{RC}^{\lambda}_{\mathrm{fix}}$ follows. □

A further generalization appears if the condition that the covering objects are of prescribed fixed type is dropped and the side lengths of rectangles are allowed to vary in a fixed closed interval.

Definition 4. *The* (2-sided) rectangular covering problem $\mathrm{RC}^{\lambda}(2)$ *is the following search problem: For a fixed real lattice constant* $\lambda > 0$*, let a point set* $M = \{z_1, \ldots, z_n\} \subset I_\lambda$ *(*$n \in \mathbb{N}$*) and* $k, k' \in \mathbb{R}_+, k' \geq k > 0$ *be given. Find a set*

R of isothetical rectangles whose side lengths lie in the closed interval $[k, k']$ such that $M \subset \bigcup_{r\in R} r \cap I_\lambda$ and $|R|$ is minimized. In the decision version $\mathrm{DRC}^\lambda(2)$ *a further input parameter $N \in \mathbb{N}$ is given, and it has to be decided whether there exists a covering R of M such that $|R| \leq N$.*

Again observing that the problems stated in Definition 2 are special cases of those just defined, namely for $t = k = k'$, we obtain immediately from Corollary 1:

Proposition 2. $\mathrm{DRC}^\lambda(2)$ *is* NP*-complete and* $\mathrm{RC}^\lambda(2)$ *is* NP*-hard.* □

The objective function posed for the problems so far is the constant $w \equiv 1$ corresponding to minimizing $w(R) = |R|$. Therefore it is natural to ask for rectangular covering problems optimizing more complex objective functions.

Definition 5. *Let w be an objective function. The* (2-sided) rectangular covering problem w.r.t. w $\mathrm{RC}^\lambda_w(2)$ *is the problem* $\mathrm{RC}^\lambda(2)$ *where (instead of $|R|$) $w(R)$ has to be minimized over all feasible coverings R of an input instance $M = \{z_1, \ldots, z_n\} \subset I_\lambda$. In the* decision version $\mathrm{DRC}^\lambda_w(2)$ *a further input parameter $W \in \mathbb{R}_+$ is given, and it has to be decided whether there exists a covering R of M such that $w(R) \leq W$.*

Notice that for a monotone objective w, all previously defined problems appear as special cases of those in Definition 5, respectively. Hence, by the previous complexity results we obtain in that case NP-completeness of $\mathrm{DRC}^\lambda_w(2)$ and NP-hardness of $\mathrm{RC}^\lambda_w(2)$ without further work. But we can derive similar complexity results for these problems also, when an arbitrary, i.e. not necessarily monotone, objective w has to be minimized.

Theorem 1. $\mathrm{DRC}^\lambda_w(2)$ *is* NP*-complete and* $\mathrm{RC}^\lambda_w(2)$ *is* NP*-hard, for an arbitrary objective function w on rectangles according to Definition 1.*

Proof. For an arbitrary objective w on rectangles, consider the problems addressed in the theorem for the special choice $\lambda = 1$. Again it is obvious that $\mathrm{DRC}^1_w(2) \in \mathrm{NP}$ since $w(r)$ is computable in constant time by definition. We show NP-completeness of $\mathrm{DRC}^1_w(2)$ by reduction from $\mathrm{DQC}^1_{\mathrm{fix}}$. Let (M, t, N) be an instance of the latter problem. From this the instance $(M, k = t, k' = t, W = Nw(q_t))$ of $\mathrm{DRC}^1_w(2)$ can be computed in polynomial time, where q_t is the square of side length t.

Assuming $(M, t, N) \in \mathrm{DQC}^1_{\mathrm{fix}}$ there is a covering Q of M such that $|Q| \leq N$. Then $(M, k = t, k' = t, W = Nw(q_t)) \in \mathrm{DRC}^1_w(2)$, since each $q \in Q$ has sides of length in $[k, k'] = \{t\}$ and moreover holds $w(Q) = |Q|w(q_t) \leq Nw(q_t) \leq W$. Conversely, if $(M, k = t, k' = t, W = Nw(q_t)) \in \mathrm{DRC}^1_w(2)$ with corresponding covering R then $(M, t, N) \in \mathrm{DQC}^1_{\mathrm{fix}}$ holds true since we have $|R| = W/w(q_t) \leq N$ rectangles (squares) of unique side length t.

In the same way $\mathrm{QC}^1_{\mathrm{fix}}$ is polynomially- and thus Turing-reducible to $\mathrm{RC}^1_w(2)$ from which follows that the latter problem is NP-hard and therefore also the more general problem $\mathrm{RC}^\lambda_w(2)$ is NP-hard. □

The following problem fixing only the left interval boundary for side lengths turns out to be also a specialization of $\mathrm{RC}^\lambda_w(2)$:

Definition 6. *Let w be an objective function for rectangles. The* (1-sided) rectangular covering problem w.r.t. w $\mathrm{RC}_w^\lambda(1)$ *is the following search problem: For a fixed real lattice constant $\lambda > 0$, let a point set $M = \{z_1, \ldots, z_n\} \subset I_\lambda$ ($n \in \mathbb{N}$) and $k \in \mathbb{R}_+, k > 0$ be given. Find a set R of isothetical rectangles each having sides of length at least k such that $M \subset \bigcup_{r \in R} r \cap I_\lambda$ and $w(R)$ is minimized. In the* decision version $\mathrm{DRC}_w^\lambda(1)$ *a further input parameter $W \in \mathbb{R}_+$ is given, and it has to be decided whether there exists a covering R of M such that $w(R) \leq W$.*

Remark 2. $\mathrm{RC}_w^\lambda(1)$ *is a special version of* $\mathrm{RC}_w^\lambda(2)$. *Indeed, as M is always finite there is a natural upper side length $k' = \max\{k, x_u(M) - x_d(M), y_u(M) - y_d(M)\} > 0$ for rectangles determined by the two extremal points $z_d(M) := (x_d(M), y_d(M)), z_u(M) := (x_u(M), y_u(M)) \in I_\lambda$. These are given by: $x_d(M) := \min_{z \in M} x(z)$, $y_d(M) := \min_{z \in M} y(z)$ and $x_u(M) := \max_{z \in M} x(z)$, $y_u(M) := \max_{z \in M} y(z)$. Notice that $z_d(M), z_u(M)$ need not be points of the input set M. This is illustrated in Fig. 1, where $z_d(S), z_u(S)$ correspond to the white dots being no members of point set $M = S$.*

There is no way to show NP-completeness of $\mathrm{DRC}_w^\lambda(1)$ simultaneously for the whole class of (monotone) objective functions w on rectangles. This can be seen considering the constant $w \equiv 1$ and an arbitrary instance (λ, M, k). In this case always one rectangle covering the whole input point set M would be an optimal solution as long as each of its sides has length at least k (cf. again Fig. 1 for $M = S$ and $0 \leq k \leq 3\lambda$, hence $y_u(S) - y_d(S) \geq k$ and $x_u(S) - x_d(S) \geq k$).

However, things may be different for other concrete monotone objective functions w underlying $\mathrm{DRC}_w^\lambda(1)$. For example, in the (from an applicational point of view interesting) case that the objective function is defined to be the sum of the area and the circumference of a rectangle plus a positive constant. Due to the variability of the lattice constant, which also is an input parameter of the problem we can prove its NP-completeness by using a tiling argument.

Definition 7. *Let* $\mathrm{RC}_{w_c}^\lambda(1)$ *(resp.* $\mathrm{DRC}_{w_c}^\lambda(1)$*) be the problem* $\mathrm{RC}_w^\lambda(1)$ *(resp.* $\mathrm{DRC}_w^\lambda(1)$*) for the objective function $w_c(r) := a(r) + u(r) + c$, where $c > 0$ is a fixed constant and $r \in \mathcal{R}$ is an admissible rectangle.*

Remark 3. *w_c is even a monotone objective function which is not hard to see.*

Theorem 2. $\mathrm{DRC}_{w_c}^\lambda(1)$ *is* NP*-complete and* $\mathrm{RC}_{w_c}^\lambda(1)$ *is* NP*-hard.*

Proof. To verify that $\mathrm{DRC}_{w_c}^\lambda(1) \in \mathrm{NP}$, let λ be fixed and let $M = \{z_1, \ldots, z_n\} \subseteq I_\lambda$ be an input set of points. For $W \in \mathbb{R}^+, k > 0$, let $R, |R| \leq |M|$, be a non-deterministically guessed set of rectangles. Let each $r \in R$ be represented by its lower left resp. upper right diagonal points $z_d(r) = (x_d(r), y_d(r)), z_u(r) = (x_u(r), y_u(r))$. For $r \in R$, compute $r \cap M$ in time $O(|M|)$ via

$r \cap M \leftarrow \emptyset$
for $i = 1$ **to** $|M|$ **do**
  if $x_d(r) \leq x_i \leq x_u(r) \wedge y_d(r) \leq y_i \leq y_u(r)$ **then**
    $r \cap M \leftarrow r \cap M \cup \{z_i\}$
end for

Simultaneously check in $O(1)$ the length condition of r posed by k and compute $w_c(r)$ in constant time. After having processed all of R we know whether $\forall r \in R : r \cap M \neq \emptyset$, whether $M \subset \bigcup_{r\in R} r \cap I_\lambda$, and also whether $w_c(R) \leq W$. Thus we are done in time $O(|M||R|)$ implying that the problem belongs to NP.

Next, we prove NP-completeness of $\mathrm{DRC}^{\lambda}_{w_c}(1)$ by reduction from $\mathrm{DQC}^{1}_{\mathrm{fix}}$. Let (M, t, N) be an instance of the latter problem, where $M = \{z_1, \ldots, z_n\}$ and $z_i = (x_i, y_i) \in \mathbb{N}^2$ $(1 \leq i \leq n \in \mathbb{N})$. From this the instance $(\lambda = t, M_t, k = t, W = Nw_c(q_t))$ of $\mathrm{DRC}^{\lambda}_{w_c}(1)$ can be computed in polynomial time, where q_t is the square of side length t and $w_c(q_t) = t^2+4t+c = const.$ Moreover M_t denotes the embedding of M into I_t which means $M = \{tz_i : 1 \leq i \leq n\}$. Assuming $(M, t, N) \in \mathrm{DQC}^{1}_{\mathrm{fix}}$ there is a covering Q of M such that $|Q| \leq N$. Then $(\lambda = t, M, k = t, W = Nw_c(q_t)) \in \mathrm{DRC}^{t}_{w_c}(1)$, since each $q \in Q$ has side length $t = k$ and moreover $w_c(Q) = \sum_{q\in Q} w_c(q) = |Q|w_c(q_t) \leq Nw_c(q_t) = W$ holds.

Conversely, if $(\lambda = t, M, k = t, W = Nw_c(q_t)) \in \mathrm{DRC}^{t}_{w_c}(1)$ with corresponding covering R, then we have $(M, t, N) \in \mathrm{DQC}^{1}_{\mathrm{fix}}$, since for each $r \in R$ having side length $\ell_j(r) > t, j \in \{x, y\}$, there exists an $n_j(r) \in \mathbb{N}$ with $\ell_j(r) = n_j(r)t$. This holds because of the embedding of M in I_t and: since rectangles are regarded as topologically closed, in an optimal covering boundary parts of rectangles can be assumed to lie on grid lines in any case. Hence, there always is a tiling of such a rectangle by $n_x(r)n_y(r)$ many squares of side length t. Since $w_c(R) \leq W$, this cannot amount to a total number of tiling objects larger than N.

In the same way $\mathrm{QC}^{1}_{\mathrm{fix}}$ is polynomially- and thus Turing-reducible to $\mathrm{RC}^{\lambda}_{w_c}(1)$ from which its NP-hardness follows. □

There is a closely related problem stemming from the application of data compression and mentioned to be NP-complete in [7]: For $n, m \in \mathbb{N}$, let $M \in \mathrm{GF}_2^{n\times m}$ be a matrix of binary entries. The associated search problem asks for a minimum cardinality set of rectangles *exactly* covering the 1-entries of M, which in a certain sense is related to the area and circumference constraints in the one-sided rectangular covering problem with objective w_c.

Finally, consider the following rectangular covering optimization problem containing a decision part. Namely, we allow only coverings R containing no more rectangles than a prescribed positive integer p:

Definition 8. *For fixed $p \in \mathbb{N}$, let $\mathrm{RC}^{\lambda}_{w}(j, \leq p)$ be the search problem $\mathrm{RC}^{\lambda}_{w}(j)$ $(j \in 1, 2)$ posed to the additional condition $|R| \leq p$ for a solution covering.*

Remark 4. *Clearly, for the constant objective $w \equiv 1$, these problems are essentially the same as the decision problems $\mathrm{DRC}^{\lambda}_{w}(j)$ $(j \in 1, 2)$ according to Definition 5 resp. Definition 6, for $N = p$.*

Due to [12] we have for its optimization variant which is defined canonically:

Theorem 3. *For each fixed $p \in \mathbb{N}$, problem $\mathrm{RC}^{\lambda}_{w_c}(1, \leq p)$ can be solved in time expressed by a polynomial in $|M|, k$ of degree $O(p)$, for an instance (λ, M, k).* □

4 Concluding Remarks

Rectangular covering problems appear in most cases to be NP-hard search problems, as far as the number of rectangles used for covering is not fixed, a priori. For designing exact algorithms the geometrical structure underlying the problem configuration should be exploited [12]. What is also needed from the point of view of numerous applications are good approximation algorithms which exist in particular for the fixed type covering problems [9]. On the other hand, there is a need of a generalization to arbitrary finite space dimensions: Modelling physical or technical systems in the framework of PDE's most often requires computations in the two-, three- or (together with a time variable) four-dimensional (discretized) physical space. But there might be also applications working in higher dimensional parameter - or configuration spaces. The problems discussed in this paper can be generalized straightforwardly to the euclidean space of (finite) dimension $d \in \mathbb{N}$. Intuitively, it is clear that the resulting problems are of a higher computational complexity than their 2-dimensional counterparts since involve some kind of further parameter d from which their NP-hardness may be derived at once. It is, however, an open problem to exactly determine the computational complexities of such higher-dimensional rectangular covering problems.

References

1. E. M. Arkin and R. Hassin, Minimum-Diameter Covering Problems, Networks 36 (2000) 147–155.
2. P. Bastian, Load Balancing for Adaptive Multigrid Methods, SIAM Journal on Scientific Computing, 19 (1998) 1303–1321.
3. S. Bespamyatnikh and M. Segal, Covering a set of points by two axis-parallel boxes, Preprint, 1999.
4. E. Boros and P. L. Hammer, On Clustering Problems with Connected Optima in Euclidean Spaces, Discrete Mathematics 75 (1989) 81–88.
5. F. C. Calheiros, A. Lucena, and C. C. de Souza, Optimal Rectangular Partitions, Networks 41 (2003) 51–67.
6. J. C. Culberson and R. A. Reckhow, Covering Polygons is Hard, Proceedings of the twenty-ninth IEEE Symposium on Foundations of Computer Science, 1988, pp. 601–611.
7. M. R. Garey and D. S. Johnson, Computers and Intractability, Freeman, New York, 1979.
8. J. Hershberger and S. Suri, Finding Tailored Partitions, Journal of Algorithms 12 (1991) 431–463.
9. D. S. Hochbaum (Ed.), Approximation Algorithms for NP-hard problems, PWS Publishing, Boston, Massachusetts, 1996.
10. M. N. Kolountzakis, On the Structure of Multiple Translational Tilings by Polygonal Regions, Discrete Comput. Geom. 23 (2000) 537–553.
11. B. Monien, E. Speckenmeyer, and O. Vornberger, Upper Bounds for Covering Problems, Methods of Operations Research 43 (1981) 419–431.
12. S. Porschen, On Covering $\mathbb{Z}$-Grid Points by Rectangles, ENDM Vol. 8, 2001.
13. S. S. Skiena, Probing Convex Polygons with Half-Planes, Journal of Algorithms 12 (1991) 359–374.

14. A. Smith, S. Suri, Rectangular Tiling in Multidimensional Arrays, Journal of Algorithms 37 (2000) 451–467.
15. S. L. Tanimoto and R. J. Fowler, Covering Image Subsets with Patches, Proceedings of the fifty-first International Conference on Pattern Recognition, 1980, pp. 835–839.

Approximating Smallest Enclosing Balls

Frank Nielsen[1] and Richard Nock[2]

[1] Sony CS Laboratories Inc., Tokyo, Japan
Frank.Nielsen@acm.org
[2] UAG-DSI-GRIMAAG, Martinique, France
Richard.Nock@martinique.univ-ag.fr

Abstract. We present two novel tailored algorithms for computing arbitrary fine approximations of the smallest enclosing ball of balls. The deterministic heuristics are based on solving relaxed decision problems using a primal-dual method.

1 Introduction

The smallest enclosing disk problem dates back to 1857 when J. J. Sylvester [20] first asked for the smallest radius disk enclosing n points on the plane. More formally, let $\text{Ball}(P,r)$ denote the ball of center P and radius r: $\text{Ball}(P,r) = \{X \in \mathbb{E}^d \mid ||PX|| \leq r\}$, where $||\cdot||$ denotes the L_2-norm of Euclidean space $\mathbb{E}^d$. Let $\mathcal{B} = \{B_1, ..., B_n\}$ be a set of n d-dimensional balls, such that $B_i = \text{Ball}(P_i, r_i)$ for $i \in \{1, ..., n\}$. Denote by $\mathcal{P}$ the ball centers $\mathcal{P} = \{P_1, ..., P_n\}$. The smallest enclosing ball of $\mathcal{B}$ is the unique ball [22], $B^* = \text{SEB}(\mathcal{B}) = \text{Ball}(C^*, r^*)$, fully enclosing $\mathcal{B}$ ($\mathcal{B} \subseteq \text{Ball}(C^*, r^*)$) of minimum radius r^*. Given a ball B, denote by $r(B)$ its radius and $C(B)$ its center. Let $x_i(P)$ denote the i-th coordinate of point P ($1 \leq i \leq d$). The smallest enclosing ball problem is also refered in the literature as the minimum enclosing ball, minimum spanning ball, minimum covering sphere, Euclidean 1-center, d-outer radius, minimum bounding sphere, or minimax problem in facility locations, etc. The smallest enclosing ball, as a fundamental primitive, finds many applications in computer graphics (collision detection, visibility culling, ...), machine learning (support vector clustering, similarity search, ...), metrology (roundness measurements, ...), facility locations (base station locations, ...), and so on. Notice that in the aforementioned applications, approximate solutions is often enough.

We survey below the main algorithms for computing the exact or approximate smallest enclosing balls. We classify previous work in Section 2 according to three algorithmic paradigms: (1) combinatorial algorithms, (2) numerical algorithms and (3) hybrid algorithms. Section 3 describes a general filtering mechanism for computing the maximum distance set-element that is then used in Section 4 to improve an implementation of a recent core-set approximation algorithm [3]. Section 5 presents a novel core-set primal-dual tailored method based on solving relaxed decision problems. Section 6 gives an alternative approach better suited for small dimensions and discusses on the algebraic degree of predicates.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 147–157, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Previous Work

Combinatorial. The smallest enclosing ball complexity was only settled in 1984 by N. Megiddo's first linear-time prune-and-search algorithm [16] for solving linear programs in fixed dimension. Later, the method was extended to the case of balls [17]. Since the smallest enclosing ball is unique and defined by at most $d+1$ support points (or balls) in strictly convex position (implying being affinely independent as well), a brute-force combinatorial algorithm requires $O_d(n^{d+2})$ time (linear memory). A major breakthrough was obtained by E. Welzl [22] who describes an elegant randomized almost tight expected $\lfloor (e-1)(d+1)! \rfloor n$ time[1] algorithm. The number of basis computations is shown to be $\tilde{O}(\log^j n)$ $(2 \leq j \leq d+1)$, so that most of the time of the algorithm is spent by checking that points/balls are inside some candidate ball.[2] For point sets being vertices of a regular simplex, the algorithm exhibits the *curse of dimensionality* as it requires $\Omega(2^d)$ recursive calls, thus limiting its tractibility up to a few dozen dimensions in practice. Recently, Chernoff-type tail bound has been given for nondegenerate input by B. Gärtner and E. Welzl [10]. Although it gives a better understanding of the power of randomization, tight worst-case bound is unknown[3] as is also the tail estimate in case of cospherical point sets. Subexponential running time was obtained by B. Gärtner [7] who described a general randomized algorithm for the class of so-called abstract optimization problems (AOP). Focusing on small instances (*i.e.*, $n = O(d)$), B. Gärtner and E. Welzl [11] presents a practical randomized approach for affinely independent points using $\tilde{O}(1.5^n)$ basis computations. T. Szabo and E. Welzl [21] further improve the bound to $\tilde{O}(1.47^n)$ using the framework of unique sink orientations of hypercubes. So far, B. Chazelle and J. Matoušek gave the current best $O(d^{O(d)}n)$ deterministic time algorithm [4]. From the pratical viewpoint, B. Gärtner [8] updated the move-to-front heuristic of E. Welzl [22] by introducing a pivot mechanism and improving the robustness of basis computations. Furthermore, K. Fischer *et al.* [6] describe a simplex-like pivoting combinatorial algorithm with a Bland-type rule that guarantees termination based on the seminal idea of T. Hopp et C. Reeve [14] of deflating an enclosing sphere: They devise a dynamic data-structure for maintaining intermediate candidate balls and a robust floating-point implementation is tested with point sets up to dimension[4] 10000. Overall complexity is $O(d^3+d^2l)$, where $l \leq \binom{n}{d+1}$ is a finite number of iterations; In practice, although the algorithm requires algebraic degree 2 on the rationals, they observe *good experimental* floating-point errors of at most 10^4 times the machine precision. For ball sets, K. Fischer and B. Gärtner show [5] that for affinely independent ball centers that E. Welzl's algorithm [22] extends to balls

[1] $e \simeq 2.71828182846...$ is the irrational number such that $\log e = 1$.

[2] This may explain why descriptions of computing the primitives were omitted in [22] since $\sum_{i=2}^{d+1}(2+\ln n)^i = o_d(n)$.

[3] That is to know the worst-case geometric configuration that implies a worst number of recursive calls (geometric realization of permutations).

[4] In fact, T. Hopp and C. Reeve [14] reported experimentally a time complexity of $\bar{O}(d^{2.3}n)$ for uniform spherical data sets.

and provide a linear programming type (LP-type) algorithm which runs in expected $\tilde{O}(2^{O(d)}n)$-time. The combinatorial algorithms described so far compute the exact smallest enclosing ball (*i.e.*, $\epsilon = 0$), report a support point/ball set and look similar to those handling linear programming. Notice that the smallest enclosing ball problem, as LP, is not known to be strongly polynomial (see P. Gritzmann and V. Klee [12] for a weakly polynomial algorithm).

Numerical. Let $d_2(\mathcal{A}, \mathcal{B})$ denote the maximum distance between all pairs (A, B) ($A \in \mathcal{A}$ and $B \in \mathcal{B}$). Observe that picking any point $P \in \mathcal{B}$ gives a 2-approximate ball $\text{Ball}(P, d_2(P, \mathcal{B}))$ (*i.e.*, $\epsilon = 1$). This allows to easily convert from relative to absolute approximation values. Motivated by computer graphics applications, J. Ritter [19] proposes a simple and fast constant approximation of the smallest enclosing ball that can be extended straightforward for points/balls in arbitrary dimension. Tight worst-case approximation ratio is unknown but can be as bad as 18.3 percents.[5] It is quite natural to state the smallest enclosing ball problem as a mathematical program. In facility locations, the smallest enclosing ball is often written as $\min_{C \in \mathbb{E}^d} F_{\mathcal{B}}(C)$ where $F_{\mathcal{B}}(X) = \max_{i \in \{1,\ldots,n\}} d_2(X, \mathcal{B})$. Since the minimum is unique, we obtain the circumcenter as $C^* = \text{argmin}_{C \in \mathbb{E}^d} F_{\mathcal{B}}(C)$. Using the ellipsoid method for solving approximately convex programs (CP), we get a $(1 + \epsilon)$-approximation in $O(d^3 n \log \frac{1}{\epsilon})$ time [13]. B. Gärtner and S. Schönherr [9] describes a generic quadratic programming (QP) solver tuned up for dense problems with few variables, as it is the case for solving basic instances. The solver *behaves polynomially* but requires arbitrary-precision linear algebra that limits its use to a few hundred dimensions. Recently, another method which turns out to perform so far best in practice, is the second-order cone programming [24] (SOCP) and requires $O(\sqrt{n} \log \frac{1}{\epsilon})$ iterations [18] using interior-point methods. Each iteration can be performed in $O(d^2(n+d))$ time for the smallest enclosing ball. G. Zhou *et al.* [24] present another algorithm, based on providing a smooth approximation of the nondifferentiable minimax function $F_{\mathcal{B}}(\cdot)$ using so-called log-exponential aggregation functions, that scale well with dn and $\frac{1}{\epsilon}$. For coarse ϵ values, say $\epsilon \in [0.001, 0.01]$, subgradient steepest-descent methods can be used as it first converges fast before slowly zigzagging towards the optimum. These numerical techniques rely on *off-the-shelves* optimization procedures that have benefited from extensive code optimization along the years but seem not particularly tuned up for the specific smallest enclosing ball problem.

Hybrid. An ϵ-core set of $\mathcal{P}$ is a subset $\mathcal{C} \subseteq \mathcal{P}$ such that the smallest enclosing ball of $\mathcal{C}$ expanded by a factor of $1+\epsilon$ fully covers set $\mathcal{P}$. Surprisingly, it was shown by M. Bădoiu *et al.* [2] that for any $\epsilon > 0$ there is a core set of size independent of dimension d. The bound was later improved to the tight $\frac{1}{\epsilon}$ value [3]. Note

[5] *E.g.*, considering a regular simplex in dimension 2. In [19], J. Ritter evaluates it to "around" 10 percents. X. Wu. [23] suggests a variant based on finding principal axis as a preprocessing stage of J. Ritter's greedy algorithm. It requires roughly twice more time and do not guarantee to perform better. (Actually, we found it experimentally worse sometimes.)

that since the smallest enclosing ball is defined by at most $d+1$ points/balls, the result is combinatorically meaningful for $\frac{1}{\epsilon} \leq d+1$. Besides, they also give a simple iterative $O(\frac{dn}{\epsilon^2})$-time algorithm (see procedure *SimpleIterativeBall* below) to compute a $(1+\epsilon)$-approximation of the smallest enclosing ball, for any $\epsilon > 0$. Combining the ellipsoid numerical approximation method with the combinatorial core-set approach yields a $O(\frac{dn}{\epsilon} + \frac{d}{\epsilon^4})$-time hybrid algorithm. P. Kumar *et al.* [15] relies on the work of [24] to obtain a better $O(\frac{dn}{\epsilon} + \frac{1}{\epsilon^{\frac{9}{2}}} \log \frac{1}{\epsilon})$-time bound.[6] S. Har-Peled mentioned an unpublished $O(\frac{dn}{\epsilon} + \frac{1}{\epsilon^2} \log^2 \frac{1}{\epsilon})$-time algorithm, so that the hybrid algorithm runs in $O(\frac{dn}{\epsilon} + \frac{1}{\epsilon^4} \log^2 n)$-time. Although not explicitly stated in the pioneer work of [2], the algorithms/bounds are still valid for ball sets (also noticed by [15]).

Our contributions. Although combinatorial algorithms exist for the smallest enclosing ball of points in very large dimensions ($d \simeq 10000$) that prove efficient in practice but lacks deterministic bound (i.e, tight worst-case analysis), we would like to emphasize on the merits of computing approximate solutions: (i) guaranteed worst-case time dependent on $\frac{1}{\epsilon}$ (the less demanding, the faster), (ii) very short code: no basis computations of at most $d+1$ points/balls are required, (iii) no special care are required for handling degeneracies (*i.e.*, cospherical points), (iv) stable: use predicates of lower degrees (see Section 6). Our contributions are summarized as follows: (i) We show an effective implementation of approximate enclosing balls of core-sets ($d \simeq 15000$ and $\epsilon \simeq 1\%$) based on distance filtering, (ii) We describe a new tailored core-set algorithm for dual decision problems, (iii) We propose an alternative effective algorithm for small dimensions, (iv) we review algorithm performances according to experiments obtained on a common platform.

3 Distance Point-Set Queries

Often, we need to compute the distance, $d_2(P, \mathcal{B})$, from a query point P to a point/ball set $\mathcal{B}$. A naive algorithm, computing distance pairs iteratively, requires $O(dn)$ time per query so that q farthest queries $d_2(\cdot, \mathcal{B})$ cost overall $O(qdn)$ time. When dimension d is large, say $d \geq 100$, computing distances of query point/set become in itself an expensive operation. Observe that $d_2(X,Y) = ||X-Y|| = \sqrt{\sum_{i=1}^{d}(X_i - Y_i)^2}$ can be written as $||X-Y||^2 = ||X||^2 + ||Y||^2 - 2 < X, Y >$, where $<,>$ denotes the vector dot product: $< X, Y >= \sum_{i=1}^{d} X_i Y_i = X^T Y$. Using Cauchy-Schwarz inequality, we have $| < X, Y > | \leq ||X||\ ||Y||$. Therefore, the distance is upper bounded by $\sqrt{||X||^2 + ||Y||^2 + 2\sqrt{||X||^2||Y||^2}} \geq ||X-Y||$. Thus when answering q farthest queries, we can first build lookup tables of $||P_i||^2$ ($P_i \in \mathcal{B}$) in a preprocessing stage in $O(dn)$ time and then use a simple distance filtering mechanism. That is, when iteratively seeking for the maximum distance given a query point X and set $\mathcal{B}$, we skip in $O(1)$ time evaluating distance

[6] More precisely, $O(\frac{dn}{\epsilon} + \frac{d^2}{\epsilon^{\frac{3}{2}}}(\frac{1}{\epsilon} + d) \log \frac{1}{\epsilon})$-time.

$d_2(X, P_i)$ if the so far maximum distance is above the upper bound given by the Cauchy-Schwarz inequality. For sets drawn from statistical distribution, let $\bar{\alpha}$ be the expected number of skipped distances, we answer q queries in $O(d(n+q)+q(1-\bar{\alpha})dn)$ time. For uniform d-cube distributions or normal distributions we observe experimentally $\bar{\alpha} \stackrel{n}{\to} 1$ (thus for $n \geq \frac{1}{\epsilon^2}$, the algorithm converges towards optimal linear $O(dn)$ time), for uniform distributions on the d-sphere, we conversely observe $\bar{\alpha} \stackrel{n}{\to} 0$. This approach extends to ball sets as well but requires extra square-root operations in order to handle ball radii.

4 Approximating Smallest Enclosing Balls of Core-Sets

Although M. Bădoiu and K. Clarkson's algorithm [3] (procedure *SimpleIterativeBall* below) extends to ball sets as well, for ease of description, we consider here point sets. The algorithm looks like *gradient-type*[7], but it is not as we noticed experimentally that the radii sequence of enclosing balls is not necessary decreasing. Given a current circumcenter, the procedure finds a farthest point of $\mathcal{B}$ to walk towards in $O(dn)$ time bypassing the costly $O(d^2n)$ time Jacobian computation required in a steepest-descent optimization. Overall cost is $O(\frac{dn}{\epsilon^2})$ time as we need to perform $\lfloor \frac{1}{\epsilon^2} \rfloor$ iterations. Using this elegant algorithm and coupling it with approximations of smallest enclosing balls of core-sets (see [2]), we obtain a $O(\frac{dn}{\epsilon} + \frac{d}{\epsilon^4})$-time algorithm (procedure *ApproximateCoreSet*). For $\frac{1}{\epsilon} = O(\sqrt[3]{n})$, the bottleneck of the algorithm is finding the core-set rather than the overall cost of simple loops.

```
1  SimpleIterativeBall(B, ε);
2  Pick arbitrary C_1 ∈ S; i ← 1;
3  a = ⌊1/ε²⌋;
4  while i ≤ a do
5      m = argmax_j ||C_i S_j|| /* Distance filtering */;
6      C_{i+1} = C_i + 1/(i+1)(S_m − C_i);
7      i ← i + 1;
8  r_a = d_2(C_a, S);
9  return Ball(C_a, r_a);
10 ApproximateCoreSet(B, ε);
11 γ = ε/3; δ = ε/3 /* Guarantee (1+δ)(1+γ) ≤ 1+ε for any ε ≤ 1 */;
12 C_1 ← {B_1}; r_1 = 0; i ← 1;
13 while d_2(C_i, B) ≥ (1+δ) r_i do
14     k = argmax_i d_2(C_i, B) /* Distance filtering */;
15     C_{i+1} ← C_i ∪ {B_k};
16     K_{i+1} ← SimpleIterativeBall(C_{i+1}, γ);
17     C_{i+1} ← C(K_{i+1}); r_{i+1} ← r(K_{i+1});
18     i ← i + 1;
19 return Ball(C_i, r_i);
```

[7] M. Bădoiu and K. Clarkson used the term gradient-like [3].

Plugging the distance filtering mechanism of Section 3, for uniform distribution of ball sets with $d \simeq 10000, n = d + 1, \epsilon \simeq 0.01$, the algorithm requires a few seconds on current commodity PCs for a mere 30-line C code. It performs better in practice than the steepest-descent method. The algorithm is adaptive according to the core-set size, bounded by $\frac{6}{\epsilon}$, but not in the iteration process of [3] as we need to loop exactly $\lfloor \frac{9}{\epsilon^2} \rfloor$ time.[8] Theoretically, this algorithm is only slightly outperformed by a SOCP solver, but its extreme simplicity coupled with the distance filtering trick make it attractive for machine learning applications.

5 Core-Sets for Decision Problems

Our novel approximation algorithms proceed by solving dual piercing *decision problems* (see Figure 1): given a set of balls $\mathcal{P} = \{B_i = \text{Ball}(P_i, r_i), i \in \{1, ..., n\}\}$ and some $r \geq 0$, determine whether $\cap\mathcal{B}(r) = \cap_{i \in \{1,...,n\}} B_i(r) = \emptyset$ or not, where $B_i(r) = \text{Ball}(P_i, r - r_i)$. We relax the 1-piercing point problem to that of a common piercing ϵr^*-ball (*i.e.*, a ball of radius ϵr^*): Namely, report whether there exists a ball $B = \text{Ball}(C, \epsilon r^*)$ such that $B \subseteq \cap\mathcal{B}(r)$ or not (see Figure 1).

Lemma. For $r \geq r^*$, there exists a ball B of radius $r(B) = r - r^*$ centered at $C(B) = C^*$ fully contained inside $\cap\mathcal{B}(r)$.

Proof. In order to ensure that C^* is in each $B_i(r)$, a sufficient condition is to have $r \geq \max_i\{r_i + d_2(P_i, C^*)\}$. Since $B_i \subseteq \text{Ball}(C^*, r^*), \forall i \in \{1, 2, ..., n\}$, we have $\max_i\{r_i + d_2(P_i, C^*)\} \leq r^* (\star)$. Thus, provided $r \geq r^*$, we have $C^* \in \cap\mathcal{B}(r)$. Now, notice that $\forall i \in \{1, 2, ..., n\}, \forall 0 \leq r' \leq (r - r_i) - d_2(P_i, C^*), \text{Ball}(C^*, r') \subseteq B_i(r)$. Thus, if we ensure that $r' \leq r - \max_i(r_i + d_2(P_i, C^*))$, then $\text{Ball}(C^*, r') \subseteq \cap\mathcal{B}(r)$. From ineq. $(\star)$, we choose $r' = r - r^*$ and obtain the lemma (see Figure 1). □

The algorithm, detailed in procedure *DecisionProblem* for point sets, builds a core-set (sets $\mathcal{C}_i$'s) iteratively for the decision problem by narrowing the feasible domain for circumcenter C^*. It is a primal-dual method since that for solving dual ball piercing problem, it requires to solve primal smallest enclosing balls.

Let k denote the maximum number of iterations of the while loop. Observe that balls B already chosen in some core-set $\mathcal{C}_i$ are necessarily pierced by points $C(K_j), j \geq i + 1$. Indeed, since $C(K_i)$ is the center of the smallest enclosing ball of the centerpoints of balls of radius r of $\mathcal{C}_i$, and $r_i = r(K_i) \leq r$, we have $d_2(C(K_i), C(B)) \leq r$ for all $B \in \mathcal{C}_i$. Moreover, since $\cap\mathcal{C}_{i+1} \subset \cap\mathcal{C}_i$ and because the smallest enclosing ball is unique, we have $r_{i+1} > r_i$. Clearly, we have $|\mathcal{C}_i| \leq 2i$. We show that k is a function depending only on d and ϵ, independent of n. Let $v_d(r)$ denote the volume of a d-dimensional ball of radius r. We have $\cap\mathcal{C}_{i+1} \subset \cap\mathcal{C}_i$ for all i. Let K_i be the unique maximal ball contained in $\cap\mathcal{C}_i$ (obtained from the smallest enclosing ball of the centers of balls contained in $\mathcal{C}_i$). If $C(K_i)$, the center of ball K_i, does not fully pierce $\mathcal{B}$, then there exists either one ball M_i or two balls M_i and N_i such that their intersection A_i (either $A_i = M_i$ or $A_i = M_i \cap N_i$) does not contain $C(K_i)$. Since A_i is convex, this means that there exists an

[8] It is of practical interest to find a better stopping criterion.

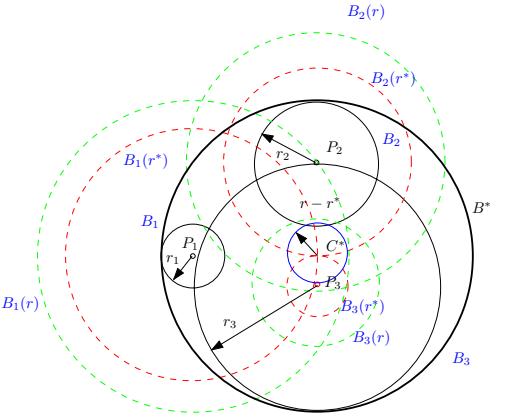


Fig. 1. Covering/piercing duality. Balls B_1, B_2, B_3 are associated to corresponding dashed balls $B_1(r), B_2(r), B_3(r)$ such that $C(B_i(r)) = P_i$ and $r(B_i(r)) = r - r_i$ for $i \in \{1, 2, 3\}$. We have $B_1(r^*) \cap B_2(r^*) \cap B_3(r^*) = \{C^*\}$. For $r \geq r^*$, there exists a ball of radius $r - r^*$ fully contained in $B_1(r) \cap B_2(r) \cap B_3(r)$.

Algorithm: *DecisionProblem*$(\mathcal{B}, \epsilon)$

Let rr be the radius obtained from a trivial 2-approximation algorithm;
Choose arbitrary $P_1 \in \mathcal{P}$; $\mathcal{C}_1 \leftarrow \{P_1\}$; $r_1 \leftarrow 0$; $i \leftarrow 1$;
while $r - r_i \geq \epsilon \frac{rr}{2}$ **do**
    Let $L_i : P_i + \lambda x_d$ /* x_d denote the unit vector of the d-th coordinate axis */;
    $\mathcal{B}_{L_i} = \{B \cap L_i \mid B \in \mathcal{B}\}$;
    if $\cap\mathcal{B}_{L_i} \neq \emptyset$ **then**
        return YES /* $r \geq r^*$ */
    else
        if $\exists B | B \cap L_i = \emptyset$ **then**
            $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{B\}$;
        else
            Let B_k and B_l such that $(B_k \cap L_i) \cap (B_l \cap L_i) = \emptyset$;
            $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{B_k, B_l\}$;
    $i \leftarrow i + 1$; $K_i = \text{SEB}(\mathcal{C}_i)$ /* Primal-Dual */ ;
    if $r(K_i) > r$ **then**
        return NO /* $r^* > r$ */
    $P_i = C(K_i)$;
return MAYBE /* $r - r^* \leq \epsilon r^*$ */ ;

hyperplane H_i separating A_i from $C(K_i)$. Let H_i' be an hyperplane parallel to H_i and passing through $C(K_i)$, $H_i'^+$ be the halfspace not containing A_i. Since $\cap\mathcal{C}_{i+1} \subset \cap\mathcal{C}_i$, we have $\text{vol}(\mathcal{C}_{i+1}) \leq \text{vol}(\mathcal{C}_i) - \frac{1}{2} v_d(r(K_i))$. Since $r(K_i) \geq \epsilon r^*$ and $\text{vol}(C_1) \leq v_d(2r^*)$, we get a sloppy upperbound $k = O(\frac{1}{\epsilon})^d$. In a good scenario, where we split in half the volume of $\cap\mathcal{C}_i$, we get $k = O(d \log_2 \frac{1}{\epsilon})$, yielding to an

overall $O(d^2 n \log_2 \frac{1}{\epsilon}) + O_{d,\epsilon}(1)$ time algorithm (improve by a factor $O(d)$ over the ellipsoid method). We observe experimentally that k tends indeed to behave as $O_d(\log \frac{1}{\epsilon})$ and that the core-set sizes are similar to the ones obtained by M. Bădoiu and K. Clarkson's algorithm. By solving $O(\log \frac{1}{\epsilon})$ decision problems, we thus obtain a $(1+\epsilon)$-approximation of the smallest enclosing ball.

6 Small Dimensions Revisited

In this section, the key difference with the previous heuristic is that dual problem sizes to solve does not depend on ϵ but are exponentially dependent on d.

Solving planar decision problems. Let $[n] = \{1, ..., n\}$ and $[x_m, x_M]$ be an interval on the x-axis where an ϵr^*-disk center might be located if it exists. (That is $x(C) \in [x_m, x_M]$ if it exists.) We initialize x_m, x_M as the x-abscissae extrema: $x_m = \max_{i\in[n]}(x_i) - r,\ x_M = \min_{i\in[n]}(x_i) + r$. If $x_M < x_m$ then clearly vertical line $L : x = \frac{x_m+x_M}{2}$ separates two extremum disks (those whose corresponding centers give rise to x_m and x_M) and therefore $\mathcal{B}(r)$ is not 1-pierceable (therefore not ϵr^*-ball pierceable). Otherwise, the algorithm proceeds by dichotomy. Let $e = \frac{x_m+x_M}{2}$ and let L denotes the vertical line $L : x = e$. Denote by $\mathcal{B}_L = \{B_i \cap L | i \in [n]\}$ the set of n y-intervals obtained as the intersection of the disks of $\mathcal{B}$ with line L. We check whether $\mathcal{B}_L = \{B_i \cap L = [a_i, b_i] | i \in [n]\}$ is 1-pierceable or not. Since $\mathcal{B}_L$ is a set of n y-intervals, we just need to check whether $\min_{i\in[n]} b_i \geq \max_{i\in[n]} a_i$ or not. If $\cap\mathcal{B}_L \neq \emptyset$, then we have found a point $(e, \min_{i\in[n]} b_i)$ in the intersection of all balls of $\mathcal{B}$ and we stop recursing. (In fact we found a $(x = e, y = [y_m = \max_i a_i, y_M = \min_i b_i])$ vertical piercing segment.) Otherwise, we have $\cap\mathcal{B}_L = \emptyset$ and need to choose on which side of L to recurse. W.l.o.g., let B_1 and B_2 denote the two disks whose corresponding y-intervals on L are disjoint. We choose to recurse on the side where $B_1 \cap B_2$ is located (if the intersection is empty then we stop by reporting the two non intersecting balls B_1 and B_2). Otherwise, $B_1 \cap B_2 \neq \emptyset$ and we branch on the side where $x_{B_1B_2} = \frac{x(C(B_1))+x(C(B_2))}{2}$ lies. At each stage of the dichotomic process, we halve the x-axis range where the solution is to be located (if it exists). We stop the recursion as soon as $x_M - x_m < \epsilon\frac{r}{2}$. Indeed, if $x_M - x_m < \epsilon\frac{r}{2}$ then we know that *no center of a ball* of radius ϵr is contained in $\cap\mathcal{B}$. (Indeed if such a ball exists then *both* $\cap\mathcal{B}_{L(x_m)} \neq \emptyset$ and $\cap\mathcal{B}_{L(x_M)} \neq \emptyset$.) Overall, we recurse at most $3 + \lceil \log_2 \frac{1}{\epsilon} \rceil$ times since the initial interval width $x_M - x_m$ is less than $2r^*$ and we consider $r \geq \frac{r^*}{2}$. Thus, by solving $O(\log_2 \frac{1}{\epsilon})$ decision problems (dichotomy search), we obtain a $O(n \log_2^2 \frac{1}{\epsilon})$-time deterministic $(1+\epsilon)$-approximation algorithm. We bootstrap this algorithm in order to get a $O(n \log_2 \frac{1}{\epsilon})$-time algorithm. The key idea is to shrink potential range $[a, b]$ of r^* by selecting iteratively different approximation ratios ϵ_i until we ensure that, at kth stage, $\epsilon_k \leq \epsilon$. Let $\text{Ball}(C, r)$ be a $(1+\epsilon)$-approximation enclosing ball. Observe that $|x(C) - x(C^*)| \leq \epsilon r^*$. We update the x-range $[x_m, x_M]$ according to the so far found piercing point abcissae $x(C)$ and current approximation factor. We start by solving the approximation of the smallest enclosing ball for $\epsilon_1 = \frac{1}{2}$. It costs $O(n \log_2 \frac{1}{\epsilon_1}) = O(n)$. Using

the final output range $[a, b]$, we now have $b - a \leq \epsilon_1 r^*$. Consider $\epsilon_2 = \frac{\epsilon_1}{2}$ and reiterate until $\epsilon_l \leq \epsilon$. The overall cost of the procedure is $\sum_{i=0}^{\lceil \log_2 \frac{1}{\epsilon} \rceil} O(n \log_2 2) = O(n \log_2 \frac{1}{\epsilon})$. The method extends to disks as well. We report on timings obtained from experiments done on 1000 trials for uniformly distributed 100000-point sets in a unit ring of width 2ϵ ($\odot$) or unit square ($\square$). Maximum (max.) and average (avg.) running times are in fractions of a second obtained by a 30-line C code on an Intel 1.6 GHz processor. (See the public code of D. E. Eberly at `http://www.magic-software.com` for a randomized implementation.)

Method/Distribution	$\square$ Square max	$\odot$ Ring max	$\square$ Square avg	$\odot$ Ring avg
D. E. Eberly ($\epsilon = 10^{-5}$)	0.7056	0.6374	0.1955	0.2767
J. Ritter [19] ($\epsilon > 0.18$)	0.0070	0.0069	0.0049	0.0049
2nd Method ($\epsilon = 10^{-2}$)	0.0343	0.0338	0.0205	0.0286
2nd Method ($\epsilon = 10^{-3}$)	0.0515	0.0444	0.0284	0.0405
2nd Method ($\epsilon = 10^{-5}$)	0.0719	0.0726	0.0473	0.0527

Predicate degree. Predicates are the basic computational atoms of algorithms that are related to their numerical stabilities. D. E. Eberly uses the *InCircle* containment predicate of algebraic degree 4 on integers ($d+2$ in dimension d for integer arithmetic. The degree drops to 2 if we consider rational arithmetic [5]). We show how to replace the predicates of algebraic degree 4 by predicates of degree 2 for integers: "Given a disk center (x_i, y_i) and a radius r_i, determine whether a point (x, y) is inside, on or outside the disk". It boils down to compute the sign of $(x-x_i)^2+(y-y_i)^2-r_i^2$. This can be achieved using another dichotomy search on line $L : x = l$. We need to ensure that if $y_m > y_M$, then there do exist two disjoint disks B_m and B_M. We regularly sample line L such that if $y_m > y_M$, then there exists a sampling point in $[y_M, y_m]$ that does not belong to both disks B_m and B_M. In order to guarantee that setting, we need to ensure some *fatness* of the intersection of $\cap\mathcal{B}(r) \cap L$ by recursing on the x-axis until we have $x_M - x_m \leq \frac{\epsilon}{\sqrt{2}}$. In that case, we know that if there was a common ϵr^*-ball intersection, then its center x-coordinate is inside $[x_m, x_M]$: this means that on L, the width of the intersection is at least $\frac{\epsilon}{\sqrt{2}}$. Therefore, a regular sampling on vertical line L with step width $\frac{\epsilon}{\sqrt{2}}$ guarantees to find a common piercing point if it exists. A straightforward implementation would yield a time complexity $O(\frac{n}{\epsilon} \log_2 \frac{1}{\epsilon})$. However, it is sufficient for each of the n disks, to find the upper most and bottom most lattice point in $O(\log_2 \frac{1}{\epsilon})$-time using the floor function. Using the bootstrapping method, we obtain a $O(n \log_2 \frac{1}{\epsilon})$ time using integer arithmetic with algebraic predicates InCircle of degree 2. In dimension 3 and higher, the dimension reduction algorithm extends with a running time $O_d(n \log_2 \frac{1}{\epsilon})$. As a side-effect, we improve the result of D. Avis and M. Houle [1] for the following problem: Given a set $\mathcal{B}$ of n d-dimensional balls of $\mathbb{E}^d$, we can find whether $\cap\mathcal{B} = \emptyset$ or report a common intersection point in $\cap\mathcal{B}$ in deterministic $O_d(n^d \log n)$ time and $O_d(n^d)$ space.

References

1. Avis D, Houle ME (1995) Computational aspects of Helly's theorem and its relatives. Int J Comp Geom Appl 5:357–367
2. Bădoiu M, Har-Peled S, Indyk P (2002) Approximate clustering via core-sets. Proc 34th IEEE Sympos Found Comput Sci (FOCS), pp 250–257. DOI 10.1145/509907.509947
3. Bădoiu M, Clarkson K (2003) Optimal core-sets for balls. Proc 14th ACM-SIAM Sympos Discrete Algorithms (SODA), pp 801–802
4. Chazelle B, Matoušek J (1996) On linear-time deterministic algorithms for optimization problems in fixed dimension. J Algorithms 21:579–597. DOI 10.1006/jagm.1996.0060
5. Fischer K, Gärtner B (2003) The smallest enclosing ball of balls: combinatorial structure and algorithms. Proc 19th ACM Sympos Comput Geom (SoCG), pp 292–301. DOI 10.1145/777792.777836
6. Fischer K, Gärtner B, Kutz M (2003) Fast smallest-enclosing-ball computation in high dimensions. Proc 11th Annu European Sympos Algorithms (ESA), LNCS 2832:630–641
7. Gärtner B (1995) A subexponential algorithm for abstract optimization problems. SIAM J Comput 24:1018–1035. DOI 10.1137/S0097539793250287
8. Gärtner B (1999) Fast and robust smallest enclosing balls. Proc 7th Annu European Sympos Algorithms (ESA), LNCS 1643:325–338
9. Gärtner B, Schönherr S (2000) An efficient, exact, and generic quadratic programming solver for geometric optimization. Proc 16th ACM Sympos Comput Geom (SoCG), pp 110–118. DOI 10.1145/336154.336191
10. Gärtner B, Welzl E (2000) On a simple sampling lemma. Electronic Notes Theor Comput Sci (eTCS), vol 31
11. Gärtner B, Welzl E (2001) Explicit and implicit enforcing: randomized optimization, Computational Discrete Mathematics (Advanced Lectures), LNCS 2122:25–46
12. Gritzmann P, Klee V (1993) Computational complexity of inner and outer j-radii of polytopes in finite-dimensional normed spaces. Mathemat Program. 59(2):163–213
13. Grötschel M, Lovasz L, Schrijver A (1993) Geometric algorithms and combinatorial optimization. Springer-Verlag
14. Hopp T, Reeve C (1996) An algorithm for computing the minimum covering sphere in any dimension. NIST 5831 Tech Rep, NIST
15. Kumar P, Mitchell JSB, Yıldırım A (2003) Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions. ACM J Exp Alg 8(1)
16. Megiddo N (1984) Linear programming in linear time when the dimension is fixed. J ACM 31(1):114–127. DOI 10.1145/2422.322418
17. Megiddo N (1989) On the ball spanned by balls. Discrete Comput Geom 4:605–610
18. Nesterov YE, Todd JE (1998) Primal-dual interior-point methods for self-scaled cones. SIAM J Optimization 8:324–364. DOI 10.1137/S1052623495290209
19. Ritter J (1990) An efficient bounding sphere. In: Glassner A (ed) Graphics Gems, pp 301–303. Academic Press
20. Sylvester JJ (1857) A question in the geometry of situation. Quarterly J Mathematics 1:79
21. Szabo T, Welzl E (2001) Unique sink orientations of cubes. Proc 42nd Ann Sympos Foundat Comp Sci (FOCS), pp 547–555
22. Welzl E (1991) Smallest enclosing disks (balls and ellipsoids). In: Maurer H (ed) New Results and New Trends in Computer Science, LNCS 555:359–370

23. Wu X (1992) A linear-time simple bounding volume algorithms. In: Kirk D (ed) Graphics Gems III, pp 301–306. Academic Press
24. Zhou G, Sun J, Toh KC (2003) Efficient algorithms for the smallest enclosing ball problem in high dimensional space. AMS Fields Institute Communications 37

Geometry Applied to Designing Spatial Structures: Joining Two Worlds

José Andrés Díaz, Reinaldo Togores, and César Otero

Dpmt. Of Geographical Engineering and Graphical Expression Techniques.
Civil Engineering Faculty. University of Cantabria. Spain

Abstract. The usefulness that Computational Geometry can reveal in the design of building and engineering structures is put forward in this article through the review and unification of the procedures for generating *C-Tangent Space Structures,* which make it possible to approximate quadric surfaces of various types, both as lattice and panel structures typologies.
A clear proposal is derived from this review: the possibility of synthesizing a great diversity of geometric design methods and techniques by means of a classic Computational Geometry construct, the *power diagram*, deriving from it the concept of *Chordal Space Structure*.

1 Definition and Typology of Space Structures

A space frame is a structural system assembled of linear elements so arranged that forces are transferred in a three-dimensional manner. In some cases, the constituent elements may be two-dimensional. Macroscopically a space frame often takes the form of a flat or curved surface [15].

That classical definition can be extended following the next classification of space structures [16]:

- *Lattice archetype*: frames composed by bars (one-dimensional elements) interconnected at nodes (zero-dimensional point objects). The structure is stabilized by the interaction of axial forces that concur at the nodes (fig.1).
- *Plate archetype*: plates (bi-dimensional elements) that conform a polyhedron's faces stabilized by the shear forces acting along its edges (one-dimensional hinges) (fig. 2 and 3).
- *Solid archetype*: structures composed by three-dimensional elements which are stabilized by the action of forces transferred between the plane facets of the solids.

2 Geometric Generation of Space Structures

The design of space structures can be approached in different ways. We now review three methods developed during the second half of the 20th century that suggest different ways for approximating a quadric surface taken as reference.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 158–167, 2004.
© Springer-Verlag Berlin Heidelberg 2004

– *Geodesic Dome*: *lattice type* structure with a configuration derived from regular or semi-regular polyhedra in which the edges are subdivides into equal number of parts ("frequency" [3]); making use of these subdivisions, a three-way grid can be induced upon the faces of the original polyhedron. The central projection of this grid's vertices on the polyhedron's *circumscribed sphere* (see fig. 1), leads to a polyhedron approximating the sphere in which only the lattice's nodes lie on the sphere's surface (more details in [7]).

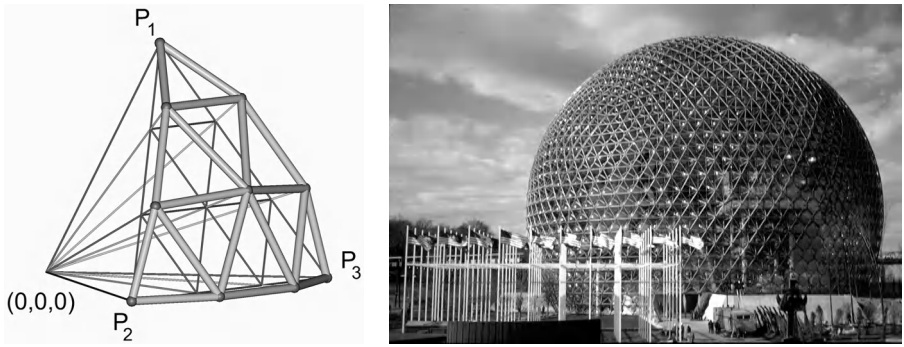


Fig. 1. *Left*: Generation of the Geodesic Dome through the projection of the three-way grid on the circumscribed sphere. *Right*: U.S. Pavilion, Montreal Universal Exposition (1967) [5]

– *Geotangent Dome*: it's a *plate type* polyhedral structure in which the edges are *tangent to a sphere*. Such a sphere is sectioned by the polyhedron's faces in such a way (fig. 2) that the faces' inscribed circles are tangent to the inscribed circles of neighboring faces. Following this rule it is possible to determine the planes containing the circles generating the polyhedron's edges from their intersection [17].

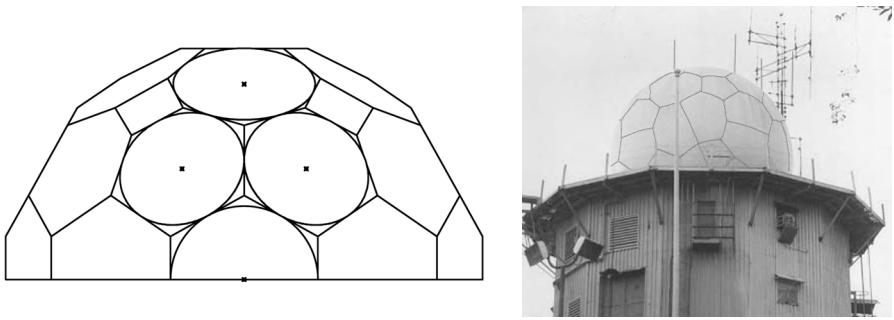


Fig. 2. Geotangent Polyhedron elevation (*left*). Nine meter diameter geotangent dome crowning Canopy Tower, Cerro Semáforo, Panamá (1963) [4]

The procedure is involved and its calculations imply the solution of a non-linear equation system through an iterative process base on successive approximations.

– *Panel Structure*: these *plate type* structures derives from lattice type geometries by applying the principle of structural and geometric *duality* (based on the concept of

a *point's polarity regarding a sphere*). Taking as a starting point the geodesic dome's circumsphere, it is possible to transform the lattice's nodes in the faces of its dual structure (fig. 3 and 5); the primitive sphere remaining as the new structure's *insphere*.

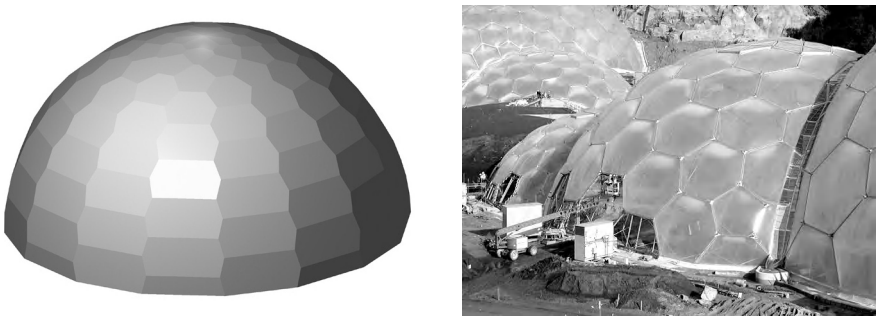


Fig. 3. Panel structure (*left*), derived as the dual polyhedron of a Schwedler type dome. (*Right*) Structures suggesting the plate typology. Eden Project, Cornwall, UK. [6]

If in this procedure the sphere on which the polarity is applied is displaced in relation to the polyhedron that is to be transformed, the resulting panel structure no longer approximates a sphere, it approximates an ellipsoid instead. The first of these procedures is known as the *Dual Transformation* (DuT), while the second is the *Dual Manipulation* (DuM) [16].

3 C-Tangent Spatial Structures

Three typologies seemingly so different as those presented in paragraph two can be integrated under a unifying proposal in the realm of Computational Geometry, through the generation of *C-Tangent structures* [11]: it is sufficient to apply to a set of points $S = \{ P_1, P_2, \ldots, P_N \}$ lying on the plane z=1 the sequence of transformations (translation, scale and inversion) which can be expressed as matrices the following way:

$$P' = [M_{TRA(-)} \cdot M_{ESC(-)} \cdot M_{INV} \cdot M_{ESC} \cdot M_{TRA}] \cdot P \quad (1)$$

followed by a projective transformation [13] (M_{HOM} matrix):

$$P'' = M_{HOM} \cdot P' \quad (2)$$

which transforms the Voronoi Diagram of these points V (S) into the polyhedral structure that approximates any quadric surface (fig. 4).

Accepting this definition for C-Tangent structures, it is feasible to perform the following interpretation of the previously defined structures:

– *Plate Structure*: a C-Tangent structure in which the proposed point set in z=1 are related by their Voronoi Diagram [11].

- *Lattice Structure*: a C-Tangent structure obtained from the z=1 point set's Delaunay triangulation [10].
- *Geotangent Structure*: a C-Tangent structure generated from the subdivision induced by the arrangement of the radical axes obtained from a tangent circles packing on z=1 [14].

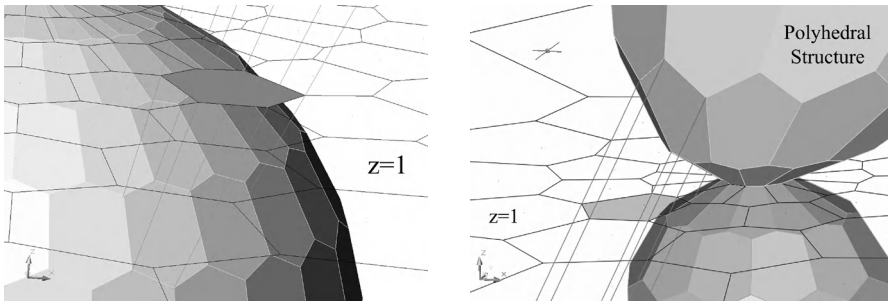


Fig. 4. Generation of C-Tangent space structures. *Left*: inversion transforms the z=1 point set's Voronoi Diagram into the polyhedral structure circumscribed to the sphere. *Right*: A projective transformation converts the approximating polyhedron into one circumscribed to a quadric

This dispersion in the starting arguments needed for the generation of C-Tangent structures is only apparent. It is enough to introduce the concept of *power diagrams* to confirm this.

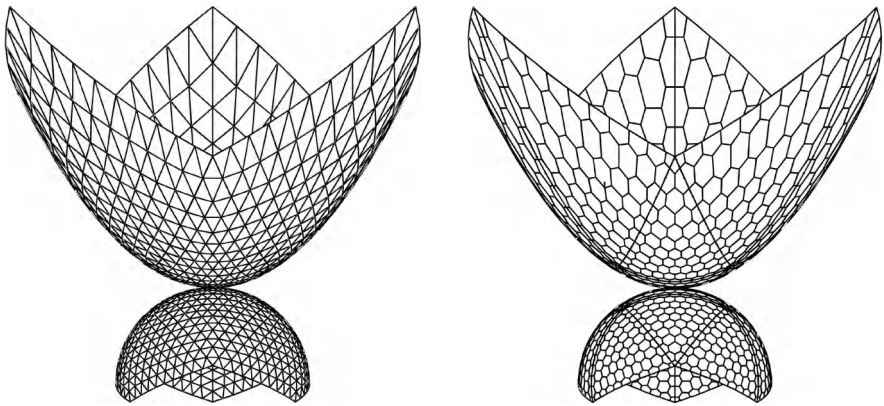


Fig. 5. Lattice mesh (*left*) and plate structure (*right*) generated from the same set of points in the z=1 plane

4 Metric and Computational Geometry Notions

4.1 Power Diagrams

From the most elemental definition:

Definition 1: the constant (signed) product of the distances from a point P to the two intersection points A and B of any line which passes through P with a circumference is called the *power of a point with respect to a circle* [12]. The power of a point P can be expressed as:

$$\text{Power} = PA \cdot PB = (d + r) \cdot (d - r) = d^2 - r^2 \tag{3}$$

where *d* is the distance from a point P to the circle's center and *r* is the circle's radius (this expression is still valid for points that lie inside the circumference).

Property 1: The locus of those points in the plane that have equal circle power with respect to two non-concentric circles is a straight line perpendicular to the line of centers. It is called *radical axis* or *power line.*

The generalization of such a definition to an n-dimensional space requires that we consider hiperspheres, not circles, centered on two generator points, in which case we formulate the locus of points in space with equal power with respect to both hiperspheres as a hiperplane orthogonal to the spheres' center line. This hiperplane is known as the *chordale* for both generator points.

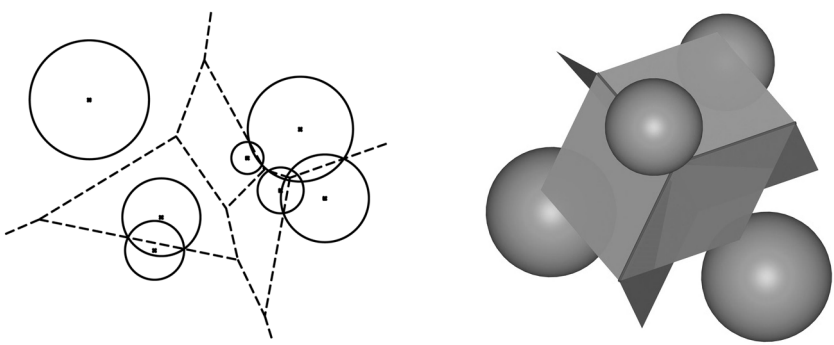


Fig. 6. Power diagrams for seven circumferences (*left*) and four spheres (*right*)

Property 2: given a collection of circumferences lying on a plane (hiperspheres in E^n) it is possible to bring about its tessellation considering nothing else than the intersection of the power lines for each pair of properly chosen neighboring circles (chordales of neighboring hiperspheres) (fig. 6). To each circumference (hipersphere), a convex region of the plane (E^n space) is associated, which is defined by the intersection of half planes (half spaces) containing those points with the least circle power. This region is known as the *power cell*, and the set of cells for the said

collection of circumferences (hiperspheres) is known as its associated *power diagram* [1].

Power diagrams and the procedures for the generation of space structures can be related through the concept of *polarity*.

4.2 Polarity in E^3

Definition 2: the *polar plane* for a point P (x_P, y_P, z_P) with respect to a quadric [8] is the locus of those points in space that are harmonic conjugates to P with respect to the two points in which any line passing through P, which is known as this *plane's pole*, intersects the given quadric (see fig. 7).

Property 3: the contact curve of the cone circumscribed to a quadric from an exterior point P is the conic section generated by the polar plane of point P. If among all the possible quadrics we select the paraboloid Ω ($z = x^2 + y^2$), it is also true that the orthogonal projection of this section on a plane z=const is a circle [12].

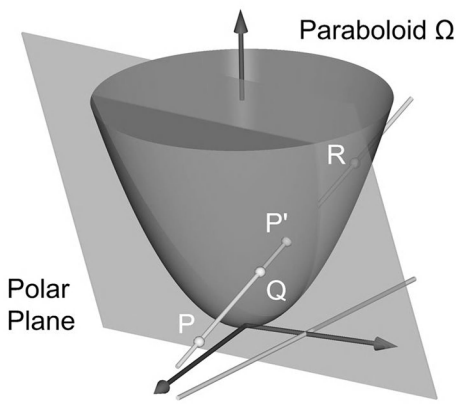


Fig. 7. Polar plane for a point P with respect to a quadric

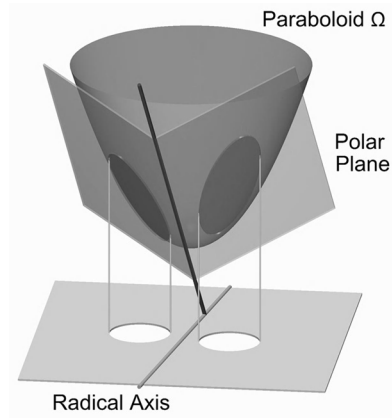


Fig. 8. Spatial interpretation of a chordale

With the projection of two of these conic sections we obtain a power diagram in which the power line is the projection on the same plane of the intersection of the polar planes containing both conic sections [1] (fig. 8).

An immediate consequence is that every power diagram is the equivalent of the orthogonal projection of the boundaries of a convex polyhedral surface (resulting from the intersection of the half-spaces defined by polar planes). This surface can be regarded as the polyhedron that approximates the quadric.

5 Revision of the Mechanism for the Definition of C-Tangent Structures. Chordal Space Structures

We propose that it is possible to generate any kind of panel structure by means of a power diagram in z=1 when it is subject to the sequence of transformations that gives rise to a C-Tangent structure. We have previously published conclusions [10], [11], [14], from studying the following two particular cases:

– A packing of tangent circumferences: in which each circumference being tangent to all its neighbors (fig. 9) (as we have seen this is the origin of *geotangent structures*).

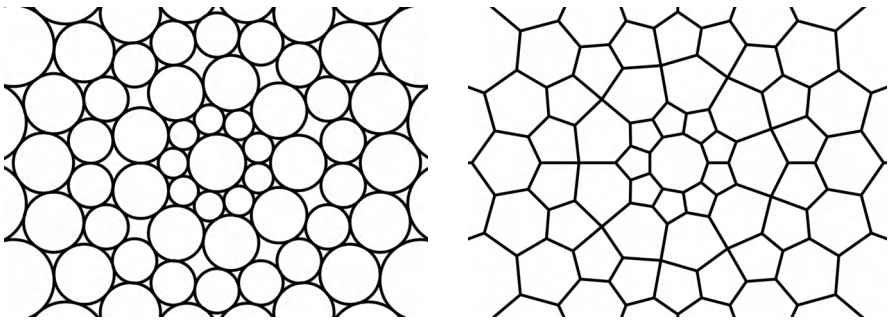


Fig. 9. A tangent circumferences packing and the planar subdivision induced by radical axes

– A subset of points lying on a plane: each point shall be considered as a zero radius circle. In this case, power lines degenerate into the perpendicular bisectors of the line segments that connect every two neighboring points, resulting in the planar subdivision that gives rise to the *Voronoi Diagram* (fig. 10) for the set of *generator points* (producing *plate type structures*).

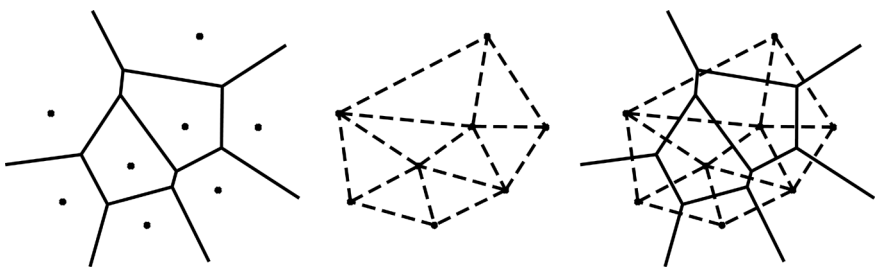


Fig. 10. Voronoi Diagram (*left*) and Delaunay Triangulation (*center*) for eight generators lying in a plane. The figure on the right shows how these structures overlap

If additionally we remember the fact that the *Voronoi Diagram* and the *Delaunay Triangulation* are dual structures, it would suffice to consider the Delaunay Triangulation for the set of zero radius circles to approximate the third of the structural typologies described: the *lattice structural type*.

For the purpose of characterizing unambiguously all the structures arising from the Power Diagrams, we propose naming them, regardless of their typology, as *chordal space structures*.

6 Generalization of the Mechanism for the Definition of Chordal Space Structures

Having to work, as stated in paragraphs three and five, with circles lying on plane z=1, could be understood as a restriction towards the problem's solution. A description of the way to overcome it follows.

Let us consider the equation of a circle expressed in its normalized form:

$$C(x, y): x^2 + y^2 - 2px - 2py + r = 0 \tag{4}$$

Completing the squares for the binomials in *x* and *y*, we obtain:

$$C(x, y): (x - p)^2 + (y - q)^2 = p^2 + q^2 - r \tag{5}$$

so that the center is the point (p, q) and the radius R is given by the formula:

$$R^2 = p^2 + q^2 - r\,. \tag{6}$$

We have set up a one-to-one correspondence between the proper circles lie in the OXY plane and the points of real Euclidean space of three dimensions (p, q, r) [12]. Points with coordinates (p, q, r) can be found to match each circle C in plane XY. According to this formulation ($C^2 \rightarrow E^3$), any collection of circles in a plane can be pictured as a cloud of points in space. And, inversely: any cloud of points in space can be understood as a collection of circles in a plane ($E^3 \rightarrow C^2$), which can be associated according to the previous section to a polyhedral surface that approximates a quadric.

Definition 3: the point that represents in the space a circle with real center but zero radius is called *point-circle* [12].

Property 4: all *point-circles* are mapped onto the points of the Ω paraboloid's surface.

Property 5: all the points above a point-circle lying on its vertical will have an associated circle in the plane with a real center (p, q) and a negative radius. We shall name such circles *virtual circles*.

7 Conclusion: Design of Chordal Space Structures

The one-to-one correspondence ($E^3 \rightarrow C^2$) defined provides us with a mechanism to associate a cloud of points in space with the faces of a polyhedron approximating paraboloid Ω, which will be partially inscribed, partially circumscribed, partially tangent to its edges and partially secant to it (fig. 11). And, thus, could also be the same to any quadric in E^3, as follows from section 3, by means of expressions (1) and (2).

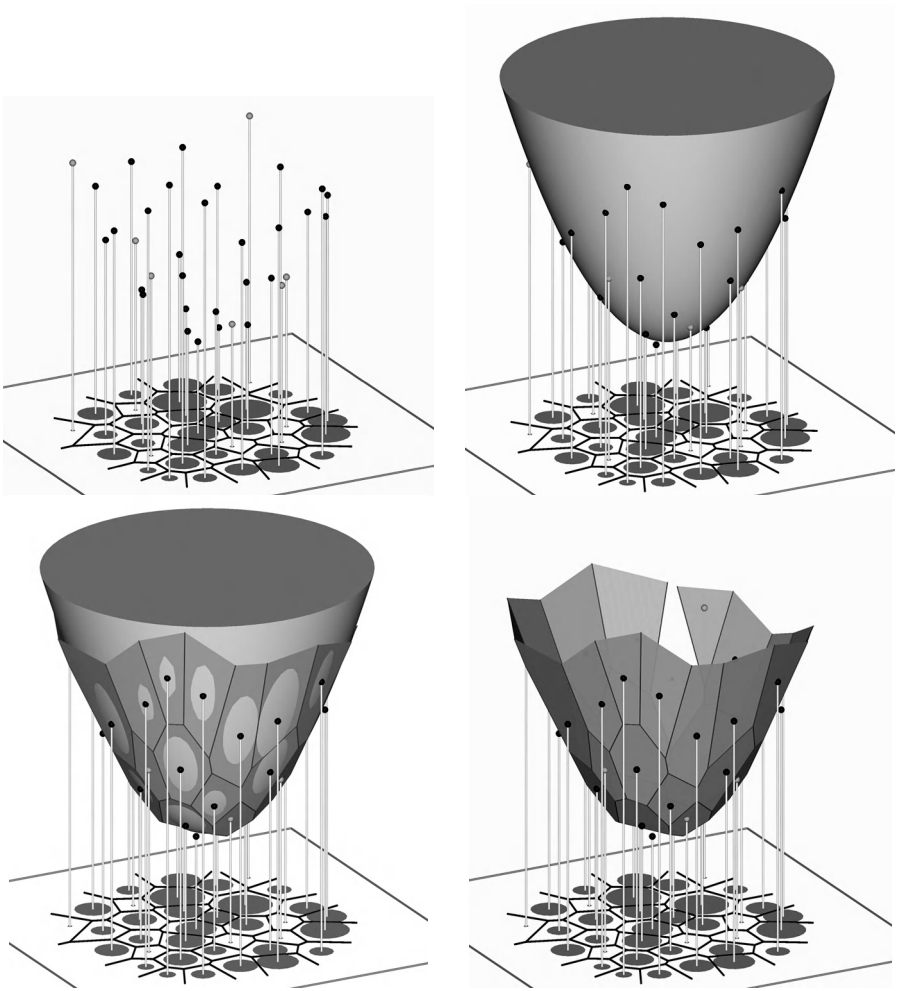


Fig. 11. A one-to-one correspondence (C2 → E3) as the mechanism of definition of the polyhedron's faces that approximate the quadric. The relative positions of points with respect to Paraboloid Ω conditions the typology of the resulting structure, which can be predicted from the associated power diagram.

The field of knowledge related with the processes by which spatial structures are obtained is plagued with innumerable typologies, procedures, classes, subclasses and patents [2], [9], that Computational Geometry can synthesize in one single category: what we have named as *Chordal Space Structures*. This proposal simplifies and widens the scope of this technical activity.

Nothing like this has been claimed before, because the intimate relation between Computational Geometry and the design of big lightweight structures remained unnoticed. Both fields are representative of progress in the XXth century and can go forward hand in hand in the XXIst.

References

1. Aurenhammer, F. (1991). Voronoi Diagrams – A survey of a fundamental geometric data structure. ACM Computing Surveys, Vol. 23, Nº. 3.
2. François Gabriel, J. (1997). Beyond the cube. The architecture of space frames and polyhedra. John Wiley and sons, New York.
3. Fuller, R. B. (1954). Building Construction. U.S. Patent 2,682,235, p. 9.
4. *http://www.canopytower.com/pub/StuTower.htm*
5. *http://www.columbia.edu/cu/gsapp/BT/DOMES/OSAKA/0425-70.jpg*
6. *http://www.eden-happenings.com/images/warm-left-ba170015.jpg*
7. Kitrick, C. J. (1990). A Unified Approach to Class I, II & III Geodesic Domes. International Journal of Space Structures, vol. 5, Nº. 3&4, pp. 223–246.
8. Mataix, C. (1947). Geometría Analítica. Dossat S. A., Madrid, pp. 191–192.
9. Nooshin, H. y Makowski, Z. S. (1990). Special Issue on Geodesic Forms. Internacional Journal of Space Structures. Vol. 5, Nº 3 y 4. Multi-Science Publishing Co. Ltd., Essex, England.
10. Otero, C., Gil, V., Álvaro J. I. (2000) CR-Tangent Meshes. IASS Journal Vol. 41 Nº. 132, pp. 41–48.
11. Otero C., Togores R. (2002). Computational Geometry and Spatial Meshes. Lecture Notes On Computer Science (Internacional Conference on Computer Science ICCS2002). Amsterdam. Vol. 2. pp. 315–324. Springer.
12. Pedoe, D. (1970). Geometry. A Comprehensive Course. Dover Publications, Inc. New York, pp. 74, 136, 138, 139.
13. Preparata, F. P., y Shamos, M. I. (1985). Computational Geometry: An Introduction. Springer-Verlag, New York, pp. 246–248.
14. Togores, R., y Otero, C. (2003). Planar Subdivisions by Radical Axes applied to Structural Morphology. Lecture Notes On Computer Science (Internacional Conference on Computer Science and its Applications, ICCSA2003), Montreal. Vol. 1, pp. 438–447, Springer.
15. Tsuboi, Y. et. al. (1984). Analisys, desing and realization of space frames. Bulletin of International Assotiation for shell and spatial structures. Working group of spatial steel structures. Nº 84-85. Volume: XXV-1/2, pp. 15.
16. Wester, T. (1990). A Geodesic Dome-Type Based on Pure Plate Action. Special Issue on Geodesic Forms. International Journal of Space Structures. Vol. 5, Nº. 3 y 4. Multi-Science Publishing Co. Ltd., Essex, England.
17. Yacoe, J. C. (1987). Polyhedral Structures that Approximate a Sphere. U. S. Patent 4,679,361.

A Robust and Fast Algorithm for Computing Exact and Approximate Shortest Visiting Routes

Håkan Jonsson

Department of Computer Science and Electrical Engineering, Luleå University of Technology, SE-971 87 Luleå, Sweden, hj@sm.luth.se

Abstract. Given a simple n-sided polygon in the plane with a boundary partitioned into subchains some of which are convex and colored, we consider the following problem: Which is the shortest route (closed path) contained in the polygon that passes through a given point on the boundary and intersects at least one vertex in each of the colored subchains? We present an optimal algorithm that solves this problem in $O(n)$ time. Previously it was known how to solve the problem optimally when each colored subchain contains one vertex only. Moreover, we show that a solution computed by the algorithm is at most a factor $\frac{2+c}{c}$ times longer than the overall shortest route that intersects the subchains (not just at vertices) if the minimal distance between vertices of different subchains is at least c times the maximal length of an edge of a subchain. Without such a bound its length can be arbitrarily longer. Furthermore, it is known that algorithms for computing such overall shortest routes suffer from numerical problems. Our algorithm is not subject to such problems.

1 Introduction

Much research has been devoted to the study of shortest paths and algorithms that compute such paths during the recent years. In many problems the shortest paths must not only go free of obstacles but also visit a set of objects [1]. Maybe the most famous example is the *Traveling Salesman Problem* (TSP) [2, 3] in which the solution is the shortest route (closed path) that visits a given set of points in the plane. In fact, the literature contains a rich mix of TSP-like problems where the path and the objects must lie within a simple polygon [4]. These include the *Zookeeper's Problem* in which the objects are convex and connected to the boundary of the simple polygon, and the shortest route (the *shortest zookeeper route*) that visits the objects must not enter their interior [5,6]. For these problems there also exists algorithms that compute provably good approximations [7,8]. However, a major drawback with many of these algorithms is that they suffer from numerical problems [9].

In this paper we present a simple linear-time algorithm for a related path-problem. The algorithm is numerically robust and computes an approximate solution to the Zookeeper's Problem. In the related problem we are given a simple n-sided polygon with a boundary partitioned into subchains, some of

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 168–177, 2004.
© Springer-Verlag Berlin Heidelberg 2004

which are convex and colored, and asked for the shortest route contained in the polygon that passes through a given point on the boundary and intersects at least one vertex in each of the colored subchains. Previously it was known how to solve the problem if each colored subchain contains one vertex only. We show that the algorithm we present here computes a solution at most a factor $\frac{2+c}{c}$ times longer than the overall shortest route that intersects the subchains (not just at vertices) if the minimal distance between vertices of different subchains is at least c times the maximal length of an edge of a subchain. Note that without such a bound the length of the solution can be arbitrarily longer and that the overall shortest route mentioned here is a shortest zookeeper route for the colored subchains.

The algorithms by Jonsson [7] and Tan [8] compute approximations at most a constant times longer than the shortest zookeeper route in all cases. However, as we discuss in Section 5, the approximation factor of the first is worse when $c \geq \frac{2}{5}$ and the second suffer from numerical problems.

In our presentation below we first outline a rather inefficient $O(n^3)$ time algorithm using dynamic programming. By observing that the parts of different shortest paths that bridge consecutive colored subchains do not intersect and by proving that the involved length functions are convex we show, in Section 3.1, how to solve the problem in $O(n^2)$ time. The latter algorithm computes and concatenates many paths and using the fact that shortest paths from points close to each other are very similar we finally improve the running time of the algorithm to $O(n)$.

2 Preliminaries

The input to the algorithm consists of an n-sided simple polygon $\mathbf{P}$ located in the Euclidean plane and a designated point s on the boundary of $\mathbf{P}$. We assume that the boundary of $\mathbf{P}$ is oriented clockwise around its interior so that locally the interior is to the right during a traversal of the boundary. The polygon is defined by the coordinates of its vertices which are stored in an array and in the order they appear along the boundary. Each edge of $\mathbf{P}$ is represented by the vertices that delimit it. The point s is referred to as the *entrance*. It is stored together with a pointer to the edge that contains it.

The input also consist of a set of m pairwise disjoint and convex subchains $\{C_1, \ldots, C_m\}$ of the boundary of $\mathbf{P}$ where the indices indicate the order in which the subchains are encountered during a clockwise scan of the boundary of $\mathbf{P}$. The subchains are indexed so that the entrance lies between C_m and C_1. The order induces a natural order also on the vertices of a subchain so that there is always a first and a last vertex. Each subchain is represented by two indices into the array of polygon vertices that point to these extreme vertices. We use $v_{i1}, v_{i2}, \ldots, v_{ik_i}$ to denote the vertices of subchain C_i indexed in clockwise order along $\mathbf{P}$, where k_i denotes the total number of vertices in C_i.

The algorithm outputs a (closed) path. We use $\pi(p, q)$ to denote the shortest path in $\mathbf{P}$ that connects p with q. The shortest path in $\mathbf{P}$ that starts at s, ends

at v_{ij}, and intersects at least one vertex in each of the subchains $C_1, C_2, \ldots, C_i$ in this given order is denoted S_{ij} and referred to as a *shortest vertex path*. Each vertex of a subchain is stored with the shortest vertex path that ends at the vertex. Rather than explicitly storing the entire path S_{ij} at v_{ij} its length from s is stored together with a pointer to the first vertex in C_{i-1} it intersects; for $i = 1$, there is no preceding subchain and the pointers instead refer to the entrance s. The rational behind this is that it reduces the storage needed to a total of $O(n)$ while it is still possible to reconstruct the actual path by tracing the pointers between subchains back to s and connecting them with shortest paths. We use S_i to denote the set of all shortest vertex paths S_{ij} for $j \in [1, \ldots, k_i]$.

3 The Algorithm

The algorithm proceeds in five steps:

1. Compute the shortest paths in $\mathbf{P}$ from s to each of the vertices of C_1. This gives us the paths S_1.
2. FOR i:= 2 TO m DO
   a) Compute S_i using $S_{(i-1)}$.
3. Compute the shortest paths in $\mathbf{P}$ from s to each of the vertices of C_m.
4. FOR i:= 1 TO k_m DO
   a) Connect the shortest path in $\mathbf{P}$ from s to the vertex v_{mi} of C_m with the shortest vertex path to that vertex.
5. Report the shortest of the paths formed in the previous step as the result.

Steps 1 and 3 can readily be solved in $O(n)$ time using *shortest path trees* [10] after $\mathbf{P}$ has been triangulated in $O(n)$ time [11]. Given the paths computed in Step 3, Steps 4 and 5 are straightforward. However, in each iteration of Step 2 there are $O(n^2)$ shortest paths between vertices in the subchains and each of these paths have size $O(n)$. We next show that only $O(n)$ of these paths need to be considered and that we can find them in $O(n)$ steps. The issue of how to perform the computation of the shortest paths efficiently is treated in Section 3.2.

3.1 The Computation in Step 2

The computation in each iteration of Step 2 takes place locally in a subpolygon of $\mathbf{P}$ bounded by shortest paths. During iteration i this subpolygon is denoted $\mathbf{R}'_i$ and defined as the union of all shortest paths between C_i and C_{i-1}. $\mathbf{R}'_i$ is bounded by two additional convex chains apart from the subchains. They are the shortest path U_i between the first vertex of C_i and the last vertex of C_{i-1}, and the shortest path L_i connecting the first vertex of C_{i-1} with the last vertex of C_i. All four chains that bound $\mathbf{R}'_i$ bulge towards the interior of $\mathbf{R}'_i$. Since the shortest path in $\mathbf{P}$ that connects a given sequence of vertices can be computed in $O(n)$ time [5], we have:

Lemma 1. *All regions $\boldsymbol{R}'_i$ can be computed in a total of $O(n)$ time.*

In each iteration of Step 2, the algorithm traverses the chain C_i one vertex at a time starting at the first vertex. For each vertex v_{ij} encountered the vertex $v_{(i-1)k}$ in C_{i-1} through which S_{ij} passes is computed by traversing the chain C_{i-1} backwards in the direction towards its first vertex. The latter traversal continues as long as the length of the shortest visiting path from v_{ij}, via the vertices considered in C_{i-1} and further on to s along one of the chains in S_{i-1}, decreases. It halts whenever an additional step would increase its length or the first vertex of C_{i-1} is reached. That this gives the vertex through which S_{ij} pass follows from the following:

Lemma 2. *The length of the shortest path from v_{ij} on C_i via a point x on C_{i-1} that visits at least one (arbitrary) point on $C_1, \ldots, C_{i-2}$ in order and ends at s is a convex function in x.*

Proof. The shortest path through x consists of two parts. One is the length of the shortest path from v_{ij} to a point x on the subchain C_{i-1}. Since C_{i-1} is convex it follows that the function is also convex (see [12, 13, 14]). The other function is the length of the shortest path $s_{i-1}(x)$ in $\mathbf{P}$ that starts at the point x, intersects at least one (arbitrary) point on $C_{i-2}, \ldots, C_1$ in this order, and ends at s. For $i = 1$, $|s_1(x)|$ is convex since — as we argued above — it is the shortest path between a single point and a convex chain. For $i > 1$ assume that $|s_{i-2}(x)|$ is convex. By the *reflection principle*[1], and the fact the a shortest path is locally optimal at each point, the parts of the shortest visiting paths $s_{i-1}(x_1)$ and $s_{i-1}(x_2)$ where $x_1 \neq x_2$ diverge between C_{i-2} and C_{i-1} [12]. From this, and the convexity of the subchains, we conclude that the length of the part of $s_{i-1}(x_1)$ between C_{i-2} and C_{i-1} is convex as well. From this the lemma follows since the sum of two convex functions is in itself a convex function.

In fact, as a consequence of Lemma 2 there could be two vertices where the (minimal) lengths of the shortest paths are equal, in which case we choose the vertex closest to the first point of C_{i-1} (along C_{i-1}). In this case the minima itself is located between the vertices[2]. When the shortest visiting path S_{ij} has been found, the traversal along C_i continues to the next vertex $v_{i(j+1)}$ and the vertex on C_{i-1} through which $S_{i(j+1)}$ passes is found (again) by traversing C_{i-1}. This is repeated until the end of C_i is reached. We have:

Lemma 3. *$S_{i(j+1)}$ intersects C_{i-1} either at the vertex where S_{ij} intersects C_{i-1} or at a vertex closer to the first vertex of C_{i-1}.*

Proof. Assume that contrary to the lemma S_{ij} intersects C_{i-1} closer to the first vertex of C_{i-1} than $S_{i(j+1)}$. This then means that the parts of S_{ij} and

[1] Attributed to Heron of Alexandria [15]. In optics, the reflection principle is also referred to as *Snell's law of reflection*, which was discovered by Willebrord van Roijen Snell[ius] in 1621 but not known until 1703 when Christiaan Huygens published Snell's result in his *Dioptrica* [16].

[2] This is also true in cases when there is but one vertex at which the length is minimized but where the minima does not coincide with the vertex.

$S_{i(j+1)}$ that go from C_i to C_{i-1} intersect. Let $v_{(i-1)k}$ be the vertex where $S_{i(j+1)}$ intersects C_{i-1} and let $v_{(i-1)(k+1)}$ be the vertex where S_{ij} intersects C_{i-1}. Then, since $S_{ij} = \pi(v_{ij}, v_{(i-1)(k+1)}) \cup S_{(i-1)(k+1)}$ is the shortest possible,

$$|\pi(v_{ij}, v_{(i-1)(k+1)})| + |S_{(i-1)(k+1)}| \leq |\pi(v_{ij}, v_{(i-1)k})| + |S_{(i-1)k}|. \quad (1)$$

It is well-known how to prove that the sum of the lengths of two opposing sides of a convex quadrilateral is less than the sum of the lengths of the diagonals. By similar reasoning, and the fact the $\mathbf{R}'_i$ is bounded by shortest paths, follows that $|\pi(v_{ij}, v_{(i-1)k})| + |\pi(v_{i(j+1)}, v_{(i-1)(k+1)})| < |\pi(v_{ij}, v_{(i-1)(k+1)})| + |\pi(v_{i(j+1)}, v_{(i-1)k})|$, which together with Eq. 1 gives us that

$$|\pi(v_{i(j+1)}, v_{(i-1)(k+1)})| + |S_{(i-1)(k+1)}| - |\pi(v_{i(j+1)}, v_{(i-1)k})| < |S_{(i-1)k}|. \quad (2)$$

But $S_{i(j+1)}$ is also a shortest possible path why $|\pi(v_{i(j+1)}, v_{(i-1)k})| + |S_{(i-1)k}| \leq |\pi(v_{i(j+1)}, v_{(i-1)(k+1)})| + |S_{(i-1)(k+1)}|$, from which we conclude that $|S_{(i-1)k}| \leq |\pi(v_{i(j+1)}, v_{(i-1)(k+1)})| + |S_{(i-1)(k+1)}| - |\pi(v_{i(j+1)}, v_{(i-1)k})|$ in contradiction to Eq. 2. From this the lemma follows.

Lemma 3 implies that there is no need to backtrack along C_{i-1}. Therefore, the traversal along C_{i-1} in one iteration of Step 2 continues from where the traversal halted in the previous iteration. Since the traversals both begin at one end of the subchains, we have:

Corollary 1. *C_i and C_{i-1} are traversed once each throughout the computation of the set S_i.*

From Corollary 1 follows that $O(n)$ shortest paths are considered throughout Step 2. Computing one of them individually can be done in $O(n)$ time [17]. However, this time complexity can be reduced by utilizing similarities between shortest paths that start and end close to each other. Below we show how to compute *all* paths in total $O(n)$ time.

3.2 Efficient Computation of Shortest Paths

During the traversals performed in Step 2 the lengths of the shortest paths between the vertices considered are needed to judge when the optimal path has been found. As mentioned one could compute the shortest paths, and hence their lengths, from scratch when needed. We take another, more efficient, approach here: The shortest path from one vertex is obtained by modifying the shortest path from the closest preceding vertex. Assuming that the shortest path from the vertex where the traversal starts has already been computed we show how the shortest path from the other vertex is computed.

It is helpful to think of the computation as the movement of the start point p of a shortest path with fixed end point q in the other subchain along the edge that connects the two vertices while the shortest path $\pi(p, q)$ is maintained. Every shortest path has a *type* which is the ordered sequence of points at which the path bends and finally ends. The type uniquely determines its path since

a shortest path is the concatenation of line segments (shortest paths) between bends. Now, imagine that p is moved and consider how the path $\pi(p, q)$ and its type change. In general, what happens is that the first edge of the shortest path changes only. The rest of the path is unaffected by the move and the type stays invariant. However, at certain points either the first and second edges become colinear, in which case they are merged into one edge and the first bend disappears from the type, or the first edge hits a vertex of $\mathbf{P}$, in which case the first edge is split in two and a new bend is introduced in the type. The points where this happens are called *event points*.

At an event point the start point and the first two bends b_1 and b_2 of the shortest path are colinear[3]. There are four closely related variants depending on where in $\mathbf{R}'_i$ the bends reside:

- Both bends lie in L_i. (Fig. 1a)
- Both bends lie in U_i. (Fig. 1b)
- Bend b_1 lie in U_i while b_2 lie in L_i. (Fig. 1c)
- Bend b_1 lie in L_i while b_2 lie in U_i. (Fig. 1d)

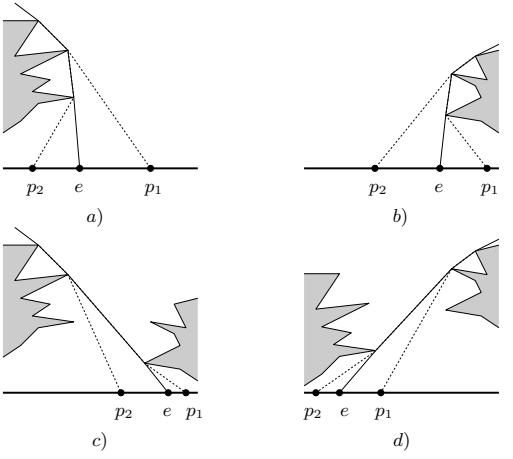


Fig. 1. The four different kinds of event points e that might occur. The shortest paths from points p_1 and p_2 in the neighborhood of e have their first bends at different points (the shaded areas show some parts of the exterior of the polygon).

The continuous movement of p is broken down into a finite number of consecutive discrete movements between neighboring event points. A move is performed by first computing the next closest event point of each variant. The point p is then moved to the closest one and the type is updated accordingly, and the next movement is performed. This is repeated until p has reached the vertex.

[3] We also consider q to be a bend.

We have informally described how one step in a traversal along an edge in one of the subchains transforms one shortest path into another by repeatedly computing event points. In fact, the union of shortest paths from event points to q is the shortest path map of q with respect to the edge along which p is moved. Recall that the shortest path map of a point on the boundary of a simple polygon is the partitioning of the polygon into regions such that the shortest paths from the point to any pair of points in a region bend at the same vertices [18]. Indeed, the procedure outlined above, where event points are computed, is an incremental construction of such a map. We still do not compute shortest path maps using the algorithms in the literature and there are two reasons for this. First of all it takes $O(n)$ time to compute a shortest path map from scratch and we need several maps which would be too costly. Second, and most important, the two traversals along C_i and C_{i-1} are performed in such a way that one end point of the shortest path is always fixed while the other is moved. Although our procedure and the algorithms for computing shortest path maps are closely related they are not the same. What we describe should be seen not as the construction of a single map rooted in q but merely as a part of the construction of a set of shortest paths between the subchains C_{i-1} and C_i during the traversals.

We now turn to the technical details on how to compute the event points efficiently. From now on we assume that p belongs to C_i and q belongs to C_{i-1}, and concentrate on the traversal along C_i; the traversal along the other subchain is carried out analogous. To compute event points we make use of two shortest paths from p. One to q and one to the first vertex $v_{(i-1)1}$ of C_{i-1}. The two shortest paths $\pi(p, q)$ and $\pi(p, v_{(i-1)1})$ are related in that $\pi(p, v_{(i-1)1})$ lie between L_i and $\pi(p, q)$

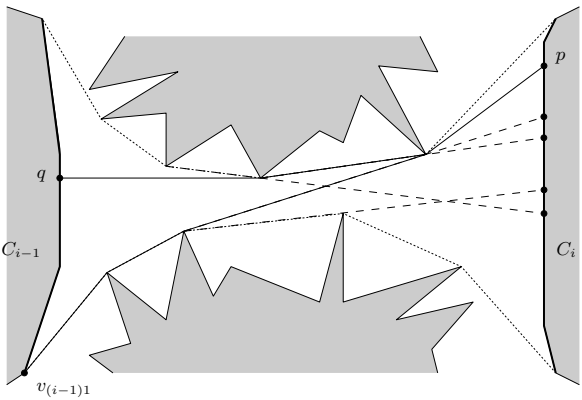


Fig. 2. The four next potential event points on C_i (the four points on C_i below the point p) when the computation of the next shortest vertex path has reached p.

During the movement of p, when event points are computed, we maintain both shortest paths by computing event points based on both paths (Fig 2). As described above this can be done by extending edges incident to the first two bends, and compute their intersection with the edge that contain p.

Lemma 4. *The set S_i can be computed in $O(|\boldsymbol{R}'_i|)$ time.*

Proof. By Corollary 1, each vertex of $\mathbf{R}'_i$ in inserted and removed from the types of the paths at most once. Moreover, since the four polygonal chains that bound $\mathbf{R}'_i$ are convex, each of the vertices also appear as first and second bends at most once each. While being such a bend they define at most two event points (either as shown in Fig. 1a and 1c or Fig. 1b and 1d), by which the lemma follows.

Then, by Lemma 1, Lemma 4, and the fact that $\sum_i |\mathbf{R}'_i|$ belongs to $O(n)$, we finally have:

Theorem 1. *Given a simple n-sided polygon with a boundary partitioned into subchains some of which are convex and colored, the shortest route (closed path) contained in the polygon that pass through a given point on the boundary and intersects at least one vertex in each of the colored subchains can be computed in $O(n)$ time.*

4 Approximating a Shortest Zookeeper Route

Our algorithm computes a path restricted to pass through vertices of the polygon. If this restriction is lifted we get a problem that has been studied extensively in the literature, namely the *Zookeeper's Problem* which asks for the shortest route in $\mathbf{P}$ that visits all convex and colored subchains in the boundary of $\mathbf{P}$[4].

It is easy to see that the route we compute can be arbitrary longer than a shortest zookeeper route in the worst case. Consider a polygon in which there is one subchain only with just one long edge and the entrance is located on some other edge of the polygon a small distance away from the mid-point of the subchain. Then the shortest zookeeper route merely follows the short path over to the mid-point (or another point on the subchain even closer) and back while the approximation is a path that goes far away to one of the vertices bounding the subchain and back again.

However, if there is a bound on the length of the edges of $\mathbf{P}$ our algorithm actually is a provably good approximation. Such bounds arise naturally in practical applications where an environment is sampled and the number of samples is much greater than the number of objects described or the objects are sampled in much more detail than the rest of the environment.

Let S_{opt} denote a shortest zookeeper route in $\mathbf{P}$ and let A_{opt} denote a solution computed by our algorithm. We then have:

[4] The original Zookeeper's Problem asks for a route that intersects a set of *disjoint convex polygons* each of which shares an edge with $\mathbf{P}$ but since the route never enters the interior of the convex polygons, the original formulation and our is the same.

Lemma 5. *Let a be the longest distance between any pair of consecutive vertices in* $\mathbf{P}$ *that belong to the same chain. Then, if the distance from any vertex in a chain to any other vertex not in the same chain is at least ca, $\frac{A_{\text{opt}}}{S_{\text{opt}}}(c) \leq \frac{2+c}{c}$.*

Proof. The path S_{opt} visits one edge per chain (possibly at one of its end points). In fact, the path consists of $m+1$ parts each of which spans between edges on two consecutive chains or connects the entrance and a subchain. To prove the lemma we show a bound on the length of the shortest path A that visits at least one end point of each of these edges compared with the length of S_{opt}. An upper bound on $|A|$ is then also an upper bound on $|A_{\text{opt}}|$ since $|A_{\text{opt}}| \leq |A|$.

Consider that part, of the $m+1$ parts, that lie between $\overline{v_{(i-1)k}v_{(i-1)(k+1)}}$ on C_{i-1} and $\overline{v_{ij}v_{i(j+1)}}$ on C_i for some $i \in [1..m+1]$, where the first and last edge are equal to s. Let S denote the part of S_{opt} that connect these edges.

S is a shortest path and therefore it lies between $\pi(v_{(i-1)k}, v_{i(j+1)})$ and $\pi(v_{(i-1)(k+1)}, v_{ij})$. Moreover, a maximum length shortest path between one of $\{v_{(i-1)k}, v_{(i-1)(k+1)}\}$ and one of $\{v_{ij}, v_{i(j+1)}\}$ follows parts of $\overline{v_{(i-1)k}v_{(i-1)(k+1)}}$, $\overline{v_{ij}v_{i(j+1)}}$, and S. The length of such a path is then at most $2a + |S|$ while $|S|$ is at least ca. The ratio $\frac{A}{S_{\text{opt}}}$ is maximized for $|S| = ca$ by which the lemma follows.

5 Numerical Robustness

It has been known for more than a decade that algorithms based on the reflection principle suffer from inherent numerical problems and all algorithms to date that compute exact solutions to the Zookeeper's Problem are no exceptions [9].

There are two previous algorithms that compute approximate zookeeper routes. The algorithm by Tan [8] achieves a better factor of approximation than the algorithm we present in this paper. Using our terminology, it computes the following points (called *images*) on the subchains: The point s_1 on C_1 that is closest to the start point s, the point s_2 on C_2 that is closest to s_1, the point s_3 on C_3 that is closest to s_2, and so on. Then the concatenation of the shortest paths connecting consecutive images is a zookeeper route at most $\sqrt{2}$ times longer than the shortest zookeeper route. However, it is not numerically robust. Consider a polygon that spirals inwards and in which the colored subchains are located such that the shortest path from each image s_{i-1} to s_i is a line segment which does not touch the boundary of the polygon. In this case, since the computed images are closest points on lines to points who are themself closest points on lines, the computed result can be expected to exhibit poor numerical accuracy. The algorithm we present in this paper and the algorithm by Jonsson [7] makes use of input data (coordinates) and intersections between lines through polygon vertices. Neither of them experience these kinds of numerical problems.

6 Conclusions

We have presented a linear-time algorithm that computes a shortest visiting route for vertices in convex subchains that are contained in the boundary of a

simple polygon. If the subchains are described in much greater detail than the rest of the polygon, or if the distances between subchains are greater than the lengths of the edges of the cages, the computed route is an approximate solution to the Zookeeper's Problem.

It would be interesting to investigate further the influence of the shape and size of the polygon and the cages on the factor of approximation. Another important and intriguing open problem is whether it is possible to compute an exact solution to the Zookeeper's Problem in linear time or not.

References

1. Mitchell, J.S.B.: Shortest paths and networks. In Goodman, J.E., O'Rourke, J., eds.: Handbook of Discrete and Computational Geometry. CRC Press LLC (1997) 445–466
2. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., eds.: The Traveling Salesman Problem. Wiley, New York, NY (1985)
3. Papadimitriou, C.H.: The Euclidean traveling salesman problem is NP-complete. Theoret. Comput. Sci. **4** (1977) 237–244
4. Jonsson, H.: The Euclidean Traveling Salesman Problem with Neighborhoods and a Connecting Fence. PhD thesis, Luleå University of Technology (2000)
5. Chin, W.P., Ntafos, S.: Optimum zookeeper routes. Info. Sci. **63** (1992) 245–259
6. Tan, X.: Shortest zookeeper's routes in simple polygons. Inform. Process. Lett. **77** (2001) 23–26
7. Jonsson, H.: An approximative solution to the Zookeeper's Problem. Information Processing Letters **87** (2003) 301–307
8. Tan, X.: Approximation algorithms for the watchman route and zookeeper's problems. Discrete Applied Mathematics **136** (2004) 363–376
9. Hershberger, J., Snoeyink, J.: An efficient solution to the zookeeper's problem. In: Proc. 6th Canad. Conf. Comput. Geom. (1994) 104–109
10. Guibas, L.J., Hershberger, J., Leven, D., Sharir, M., Tarjan, R.E.: Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica **2** (1987) 209–233
11. Chazelle, B.: Triangulating a simple polygon in linear time. Discrete Comput. Geom. **6** (1991) 485–524
12. Bespamyatnikh, S.: An O($n$log$n$) algorithm for the Zoo-keeper's problem. Comput. Geom. Theory Appl. **24** (2002) 63–74
13. Guibas, L.J., Hershberger, J.: Optimal shortest path queries in a simple polygon. J. Comput. Syst. Sci. **39** (1989) 126–152
14. Hershberger, J.: A new data structure for shortest path queries in a simple polygon. Inform. Process. Lett. **38** (1991) 231–235
15. Toussaint, G.T.: Special issue on computational geometry. In: Proceedings of the IEEE. (1992) 1347–1363
16. Sabra, A.I.: Theories of Light from Descartes to Newton. Oldbourne, London (1967)
17. Lee, D.T., Preparata, F.P.: Euclidean shortest paths in the presence of rectilinear barriers. Networks **14** (1984) 393–410
18. Hershberger, J.: An optimal visibility graph algorithm for triangulated simple polygons. Algorithmica **4** (1989) 141–155

Automated Model Generation System Based on Freeform Deformation and Genetic Algorithm

Hyunpung Park and Kwan H. Lee

Department of Mechatronics, Kwangju Institute of Science and Technology,
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Republic of Korea
{baram, lee}@kyebek.kjist.ac.kr
http://kyebek9.kjist.ac.kr

Abstract. In this paper, we propose an automated model generation system that assists the user's creativity in conceptual design. The system focuses on creating various modified versions of an existing model, namely a mesh model. Since it is difficult to control mesh models parametrically, we developed a parametric control method that controls the object shape indirectly by using a control mesh. A new model is obtained by deforming an object model. Generated models are evolved, taking into account the user's preference, by using genetic algorithms. The main topics of this paper are 1) automated construction of a control mesh, 2) management of geometric constraints, and 3) evolution of generated models. We applied our proposed system to a car model and the generated new models are shown in the example.

1 Introduction

Conceptual modeling forms the basis in the development of any new product models. Many researches have been performed to facilitate conceptual modeling. A genetic algorithm is one of the powerful techniques applied to assist conceptual modeling because it has both evolutionary and creative factors [1].

There are two factors to be considered in conceptual modeling: engineering and aesthetic factors. In engineering factors, the shape of a model should satisfy engineering requirements such as strength, noise, and material cost. Most of the developed systems focus on these engineering factors [2][3][4].

A few researches deal with aesthetic factors. Among those researches, the work of Nishino [5] is notable. A model is represented by a set of implicit surfaces. Parameters for the implicit surfaces are generated and evolved by applying genetic algorithms. Users evaluate new models according to their preferences. However, deformed shapes are so arbitrary that many meaningless shapes are generated.

In this paper, we propose an automated model generation system based on freeform deformation (FFD) and genetic algorithms to produce more reasonable shapes. We assume that the object models are industrial products with aesthetic shapes and represented by polyhedral meshes. This assumption is reasonable because the use of mesh models in aesthetic shapes is currently increasing.

In order to automate model generation, it is necessary to represent a model in a parametric form. In solid or surface models, they are represented parametrically due

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 178–187, 2004.
© Springer-Verlag Berlin Heidelberg 2004

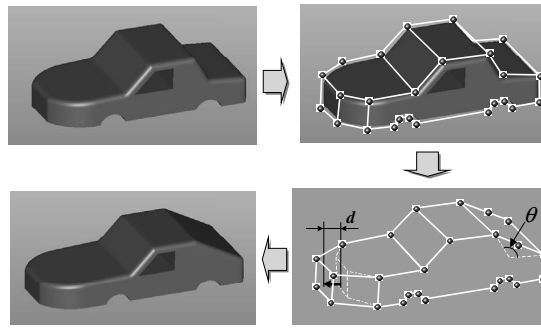


Fig. 1. A parametric control method for mesh models

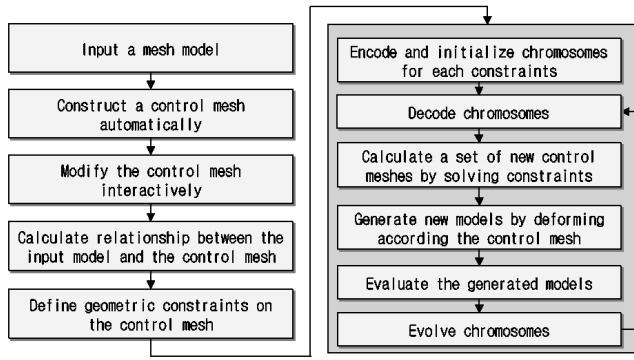


Fig. 2. Process for automated model generation

to their characteristics. However, there is no way to represent mesh models in a parametric form. In order to solve this problem, we propose a new control method.

In the proposed method, parametric control for a mesh model is achieved by utilizing a control mesh. A control mesh is a closed polygonal mesh satisfying the condition that the deviation between an object model and a control mesh is less than a given tolerance. Geometric constraints for parametric control are imposed on the control mesh. The control mesh is modified by changing the parameter values and the inside model is modified using an existing deformation technique according to the control mesh. Figure 1 shows the conceptual procedure for controlling a mesh model parametrically (refer to [6][7][8][9] for more information for freeform deformation).

The overall procedure for the proposed system is shown in figure 2. For a given object, the system calculates a control mesh automatically. A user may modify it to reflect the user's design objectives. Then the user imposes geometric constraints on the control mesh. Parameters for each constraint are generated and evolved by genetic algorithms. In each generation in a genetic algorithm, a new control mesh is calculated by solving the constraints, and a new shape is generated by FFD according to the new control mesh. The user evaluates the generated models and the process is repeated. The user may select some of the generated models and use them for further modeling.

The paper mainly focuses on three topics: 1) algorithms to automatically generate a control mesh for a given model, 2) management of geometric constraints that are appropriate to parametric control, and 3) applying genetic algorithms for model generation. A detailed explanation for each topic is given in the following sections.

2 Automated Generation of a Control Mesh

2.1 Overview

Since we control the shape of a given model by using a control mesh, the shape of the control mesh should be similar to the object model and simple enough to impose geometric constraints as well. In other words, we have to construct a control mesh as simple as possible on the condition that the deviation between the control mesh and the object model cannot exceed a given tolerance.

In the proposed algorithms to create a control mesh, an object model is projected along each axial direction in an appropriate axis system. Then, 2D control polygons are calculated for the contours of the projected shapes considering the distance tolerance. Finally, a 3D control mesh is built up by combining the three 2D control polygons. The overall procedure is as follows:

1) Determine the optimal axis system
2) Project the object model along each axis direction
3) Extract contours from each projected shape
4) Calculate 2D control polygons for each contour
5) Extrude the 2D control polygons and intersect the extruded parts

2.2 Projection and Contour Extraction

In engineering drawings, three orthogonal projection views are standard method to represent a model. By applying the same idea, each model can have any axis system where the best 3-view projection is obtained. Experimental results show that a minimum-volume bounding box (MVB) gives a good axis system in most cases (refer to [9]).

After determining an appropriate axis system, the object model is projected respectively onto three planes, *XY*, *YZ*, and *ZX* plane. In order to create a 2D control polygon, we have to extract contours from the projected shapes. In this research, an image-based approach is used for contour extraction.

The projection plane is regarded as a collection of small grids. And the projected shape is put into the grids as if an object is rasterized in a buffer to be displayed in computer graphics. For the rasterized object, the contour grids are extracted by finding connected outmost grids. A contour is completed by connecting the center points of individual contour grids. Extracted contours may have stair-step effects since the projected shape is approximated by grids, therefore, an averaging method is used to smoothen the extracted contours.

2.3 Calculation of 2D Control Polygons

A 2D control polygon for a 2D profile is a polygon that satisfies the condition that the deviation between the profile and the control polygon is less than a certain tolerance. In conceptual meaning, a 2D control polygon corresponds to a 3D control mesh.

In this step, 2D control polygons are calculated from contours extracted in the previous step. The input contour is a polygon with dense vertices. Most of the industrial products have freeform surfaces as well as primitive elements. Therefore, at first, we detect line segments in the contour. By making the set of line segments closed, we get an initial polygon. Then by calculating the distance between the contour and the initial polygon, we can identify the contour vertices that are out of the tolerance zone. These vertices belong to curved segments in the contour. The edge that is nearest to these vertices in the control polygon is removed and new edges are added by inserting the farthest vertices that are out of the tolerance zone. Until the difference between the control polygon and the contour is less than the given distance tolerance, these steps are repeated. Figure 3 illustrates the procedure of calculating a 2D control polygon.

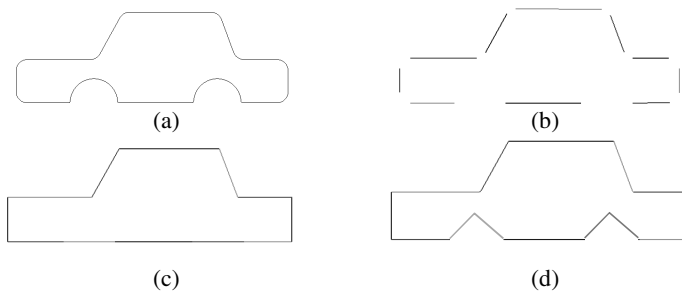


Fig. 3. Procedure of calculating 2D control polygon (a) contour data (b) line detection (c) a closed polygon (d) the final control polygon after processing the curved area

2.4 Creating 3D Control Mesh

A 3D control mesh is finally obtained by combining three 2D control polygons. Each 2D control polygon is extruded along the normal direction of the plane where the polygon lies. Then, a 3D control mesh is obtained by intersecting the three extruded parts. This is a simple solid modeling operation.

Figure 4 illustrates the procedure of constructing a control mesh from control polygons. The second column in figure 4 shows 2D control polygons for projected shapes. Parts extruded from the 2D control polygons are shown in the third column. The final 3D control mesh appears in the last column in figure 4.

Since modeling results should reflect the user's intention, the automatically generated control mesh can be modified to meet the user's demands. Therefore, the developed system provides efficient user interfaces that help to interactively modify the control mesh.

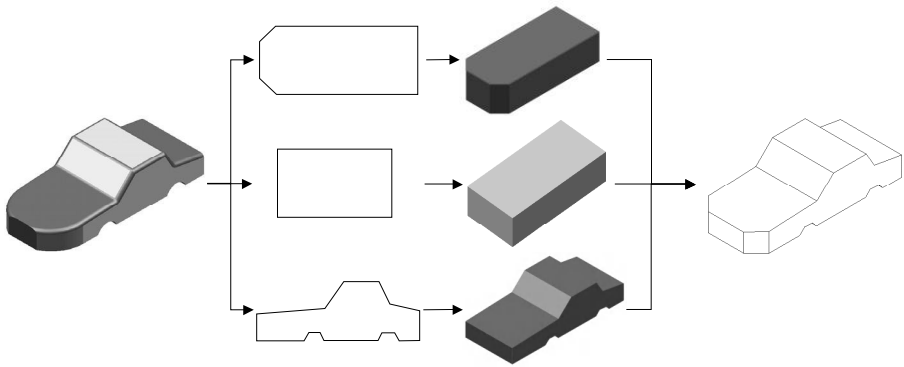


Fig. 4. Illustration of constructing a control mesh

3 Imposing Geometric Constraints

Parametric modeling, in general, constraints are defined by the relative relationship between elements [10]. We refer to this type of constraints as *relative constraints.* In addition to the relative constraints, we propose *absolute constraints* that define the absolute displacement of an element. In some cases, absolute constraints need fewer constraints than relative constraints for shape control. Figure 5 illustrates the concepts of absolute and relative constraints. In order to move the uppermost edge along normal direction, the relative constraint method requires constraints such that *θ1, d1* and (*θ1=θ2*, *d1=d2*). In contrast, only one vector, *M1,* is needed for the same purpose in the absolute constraint method.

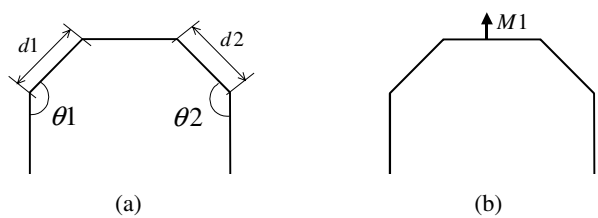


Fig. 5. Types of constraints (a) relative constraints (b) absolute constraints

Both relative and absolute constraints can be used in the proposed method. Since relative constraints are generally used, the absolute constraint method is explained below. Constraints are imposed on the faces and the edges of a control mesh. Possible absolute constraints that can be imposed on the face include parallel movement, scaling about the center point and the angles for the other faces. Absolute constraints for edges are parallel movement, and scaling. In addition, defining an equation of parameter values is also possible, as in the relative constraint method does. Vertices that do not have constraints are fixed at their initial positions. Examples of syntax expressing constraints are listed below.

ES (edge index, scaling value): Edge scaling
EM (edge index, moving direction, displacement): Edge movement
FS (face index, scaling value): Face scaling
FMN (face index, displacement): Face movement along a normal vector
FMD (face index, direction, displacement): Face movement along a given vector
AF (face index, face index, angle value): Angle between two faces
DF (face index, face index, distance): Distance between two faces

When new parameter values are given, new coordinates for control vertices are calculated by solving constraints. Since absolute and relative constraints are used in the same control mesh, the policy of solving constraints is needed. When both relative and absolute constraints are given in the same vertex, relative constraints are solved first before dealing with the absolute constraints.

4 Genetic Algorithms for Model Generation

4.1 Evolution Strategy

As mentioned in introduction, the evolution of generated models is performed by a genetic algorithm. There are many types of genetic algorithms. In optimization problems, the type of genetic algorithm applied is critical for the accuracy of the result and the convergence rate. However, in the creative design, it is difficult to define these criteria. Therefore, we apply a simple genetic algorithm. Following is the configuration of the genetic algorithm used in our system.

- Binary-encoded chromosomes with the fixed length
- Roulette wheel selection
- Single point crossover
- Mutation: change randomly chosen bits
- No substitution of chromosomes

4.2 Encoding and Decoding Geometric Constraints

The encoding scheme for geometric constraints is illustrated in figure 6. The first half of a chromosome represents the parameters of face constraints. Parameters for edge constraints are encoded into the rest of the chromosome. The length of a gene for each parameter value is calculated by using Equation (1).

$$2^{m_j-1} < (\max_j - \min_j) \times 10^{r_j} < 2^{m_j} - 1 \tag{1}$$

In the equation, r_j represents the required resolution after a decimal point and m_j is the length of a gene for the parameter. Minimum and maximum values for each constraint parameter are given by the user. Most of the constraints except *edge movement* have only one parameter. The parameters of the edge-movement constraint are an angle, θ, and a moving distance since we restrict the moving direction to the normal direction to the edge.

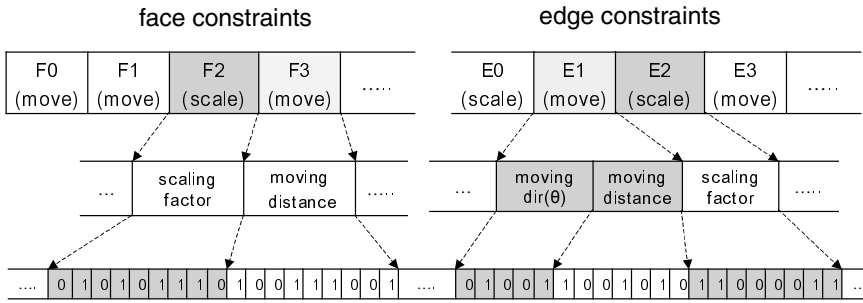


Fig. 6. Encoding geometric constraints

For each gene, the parameter value, x_j, is calculated by the following equation:

$$x_j = \min_j + decimal(substring_j) \times \frac{\max_j - \min_j}{2^{m_j} - 1} \tag{2}$$

4.3 Evaluation of the Generated Models

By decoding a chromosome, the parameter values for all the constraints are obtained. Then, the coordinates of vertices in a control mesh are calculated by solving the constraints with the obtained parameters. By deforming the original model according to the new control mesh, new models are generated. The deformation method that is used in our system is t-FFD [7].

Unlike an engineering design, the aesthetic sense of human beings cannot be represented quantitatively. Therefore, the evaluation for generated models is done by the user. For each model, the user gives a preference value from 1 to 5. A fitness value for each chromosome is calculated by the following equation:

$$v_i = \frac{p_i}{\sum_i p_i} \tag{3}$$

In the equation (3), v_i and p_i represent a fitness value and a preference value for *i-th* chromosome respectively.

5 Application Examples

Figure 7 shows the screenshot of the proposed system. The system has three menus: construction of a control mesh, management of constraints, and design evolution. Figure 7 shows a dialogue box in the menu of design evolution. It is implemented in Visual C++ V.7 using OpenGL graphics library on Windows XP platform. The hardware configuration for the system used is 1.6 GHz CPU with 512 MB memory.

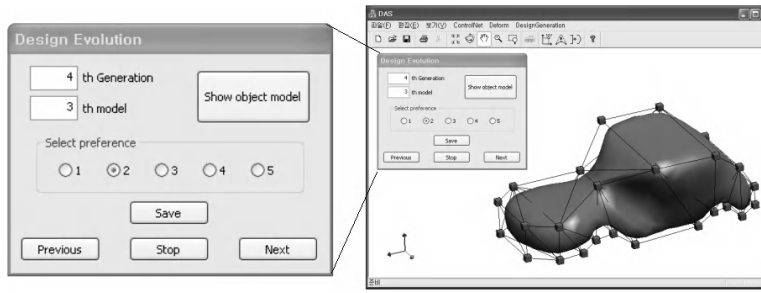
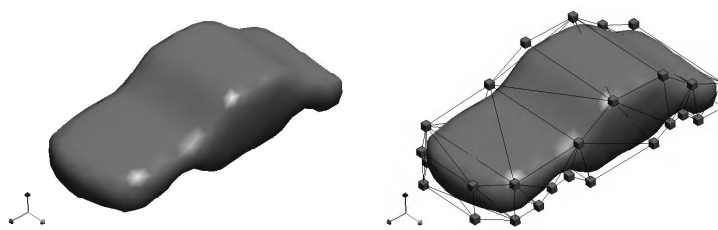


Fig. 7. Screenshot of the proposed system

We applied the proposed system to a car model shown in figure 8(a). A control mesh for the car model is illustrated in figure 8(b). It was interactively modified after being automatically generated by the algorithms described in section 2. The control mesh is in the form of a triangular mesh model since t-FFD allows only a triangular control mesh. Other types of meshes can be used in case of using other deformation methods.

In t-FFD, the deformed area is affected by the size of each triangle in the control mesh. Therefore, in order to obtain local deformation, we added the step of an implicit subdivision of a control mesh during the deformation process. That is, parameterization is done both for the object model and the subdivided control mesh. The subdivision result is hidden in the system. When the object model is deformed according to a modified control mesh, the modified control mesh is subdivided again implicitly and then, the new position of each vertex in the object is calculated by applying the parameterization result.



(a) (b)

Fig. 8. An object model and its control mesh

In order to evolve the given model, the following parameters are used in the genetic algorithm.

- Population size: 8
- Probability of crossover: 0.25
- Probability of mutation: 0.01

Since a small number of constraints are used in the example, the population size is only eight. The more constraints exist, the larger the population size may need to be. Figure 9 shows evolved models. The generated models in the first and the fourth

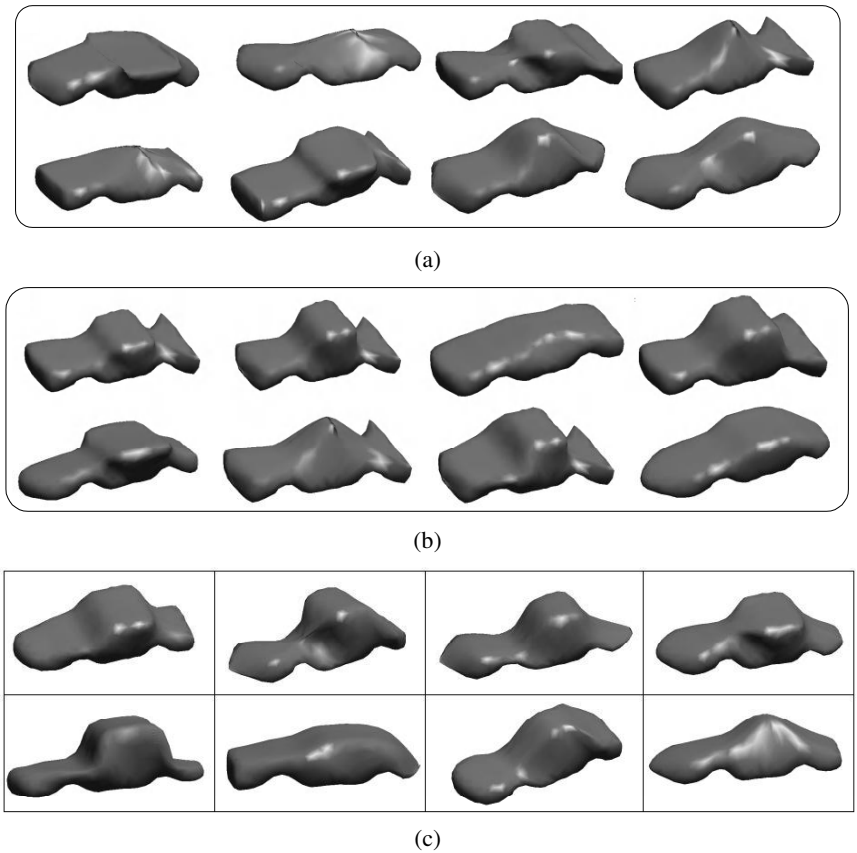


Fig. 9. Generated models (a) The first generation (b) The fourth generation (c) a collection of models selected by a user

generation are shown in figure 9(a) and figure 9(b) respectively. When evaluating the generated models, the user can keep his/her preferred models in files. Figure 9(c) shows the models selected by the user. The user can continue modeling with the selected models or can be inspired to create a new model from the generated models.

6 Conclusions

In this paper, we proposed an automated model generation system based on freeform deformation and genetic algorithms. In order to control a mesh model parametrically, a shape control method by using a control mesh was developed. The algorithms of automated generation of a control mesh will help users to create a control mesh that fits to their purpose. The concept of absolute constraints was also introduced for efficient manipulation. We applied the proposed system to a car model and showed automatically generated new models in the examples.

The automatically generated models can be used for further modeling processes or can motivate new designs. The system is expected to reduce a significant amount of time and efforts put in the early stages of product development process.

The proposed system deals only with the aesthetic factors in modeling. Therefore, in order to incorporate engineering factors, a more comprehensive analysis, for the relationship between the object model and the control mesh, should be considered. Automatically generated control meshes require user's interactions. Therefore, control mesh generation algorithms should be improved to minimize user's interactions for a complex model such as the one with inner holes.

Acknowledgement. This work was supported in part by the Ministry of Information and Communication(MIC) through the Realistic Broadcasting Research Center at K-JIST.

References

1. Rennera, G., Eka´rt, A.: Genetic algorithms in computer aided design. Computer-Aided Design, Vol. 35 (2003) 709–726
2. Qiu, S. L., Fok, S. C., Chen, C. H., Xu, S.: Conceptual Design Using Evolution Strategy. Int. J. Adv. Manuf. Technol. Vol. 20 (2002) 683–691
3. Sato, T., Hagiwara, M.: IDSET: Interactive Design System using Evolutionary Techniques. Computer-Aided Design, Vol. 33 (2001) 367–377
4. Bentley, P. J.: Generic Evolutionary Design of Solid Objects using a Genetic Algorithm. Ph.D. Thesis, Division of Computing and Control Systems, Department of Engineering, University of Huddersfield. (1996)
5. Nishino, H., Utsumiya, K., Takagi, H., Cho, S.: A 3D Modeling System for Creative Design. The 15th International Conference on Information Networking (ICOIN'01) (2001) 479–486
6. Sederberg, T. W., Parry, S. R.: Free-Form Deformation of Solid Geometric Models. SIGGRAPH '86 (1986) 151–160
7. Kobayashi, K. G., Ootsubo, K.: Deformations & shaping: t-FFD: free-form deformation by using triangular mesh. The eighth ACM symposium on Solid modeling and applications (2003) 226–234
8. Shao, J., Zhao, Y., Feng, J., Jin, X., Peng, Q.: Free-Form Deformation by using Arbitrary Topological Mesh. Proceedings of CAD & Computer Graphics 2003 (2003) 277–282
9. Ono, Y., Chen, B. Y., Nishita, T., Feng, J.: Free-Form Deformation with Automatically Generated Multiresolution Lattices. Proceedings of IEEE 2002 International Conference on Cyber Worlds (2002) 472–490
10. Anderl, R., Mendgen, R.: Parametric design and its impact on solid modeling applications. Proceedings of the third ACM symposium on Solid modeling and applications (1995) 1-12

Speculative Parallelization of a Randomized Incremental Convex Hull Algorithm*

Marcelo Cintra[1], Diego R. Llanos[2], and Belén Palop[2]

[1] School of Informatics, University of Edinburgh, Edinburgh, UK,
mc@inf.ed.ac.uk
[2] Departamento de Informática, Universidad de Valladolid, Valladolid, Spain,
{diego|bpalop}@infor.uva.es

Abstract. Finding the fastest algorithm to solve a problem is one of the main issues in Computational Geometry. Focusing only on worst case analysis or asymptotic computations leads to the development of complex data structures or hard to implement algorithms. Randomized algorithms appear in this scenario as a very useful tool in order to obtain easier implementations within a good expected time bound. However, parallel implementations of these algorithms are hard to develop and require an in-depth understanding of the language, the compiler and the underlying parallel computer architecture. In this paper we show how we can use speculative parallelization techniques to execute in parallel iterative algorithms such as randomized incremental constructions. In this paper we focus on the convex hull problem, and show that, using our speculative parallelization engine, the sequential algorithm can be automatically executed in parallel, obtaining speedups with as little as four processors, and reaching $5.15x$ speedup with 28 processors.

1 Introduction

Finding the fastest algorithm to solve a problem is one of the main issues in Computational Geometry. Focusing only on worst case analysis or asymptotic computations leads to the development of complex data structures or hard to implement algorithms. Randomized algorithms appear in this scenario as a very useful tool in order to obtain easier implementations, taking advantage of the remarkable fact that, if we study how the complexity of the algorithm is related with the ordering in which points are processed, only a tiny percentage of the orderings leads to worst case situations.

While sequential implementations of these algorithms lead to good results in terms of complexity, obtaining a parallel version is not straightforward. Sometimes the development of a sequential implementation can be accomplished without much effort, but a parallel implementation of a given incremental algorithm

* The first author has been partially supported by EPSRC under grant GR/R65169/01. The first and second authors have been partially supported by the European Commission under grant HPRI-CT-1999-00026. The third author has been partially supported by MCYT TIC2003-08933-C02-01.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 188–197, 2004.
© Springer-Verlag Berlin Heidelberg 2004

is hard to develop, requiring an in-depth understanding of the programming language, the compiler, the parallel tool set and the underlying parallel computing architecture.

In this paper we show how we can use speculative parallelization techniques to automatically parallelize sequential, incremental algorithms with a small number of dependences between iterations, such as randomized incremental constructions. In this paper we focus on the convex hull problem. Using our speculative engine, the algorithm can be automatically parallelized, with only a tiny fraction of the effort needed to design, analyze and program a parallel version (see e.g., [8]).

After analyzing the effect of input set sizes and different shapes of the data distribution, our results show that the speculative version of the sequential algorithm leads to speedups with as little as four processors, reaching a maximum of $5.15x$ for 28 processors.

The rest of the paper is organized as follows. Section 2 describes the randomized planar convex hull problem. Section 3 introduces speculative parallelization and shows how it may be used to easily obtain a parallel version of an iterative algorithm. Section 4 describes the parallel execution environment and discusses the experimental results, while section 5 concludes the paper.

2 Randomized Planar Convex Hull

Given a set S of n points in the plane, the *convex hull* of S, $CH(S)$, is the smallest convex region containing all points in S. We will use $CH(S)$ for the ordered sequence of vertices of the convex region, which are known to be points of S.

Since 1972 when Graham [9] gave the first algorithm to compute the convex hull of a set of points in $O(n \log n)$ time and $O(n)$ space, a lot of effort has been done to find algorithms reaching better lower bounds in time and space. In 1986 Kirkpatrick and Seidel [11] proved the time lower bound of $\Omega(n \log h)$, where h is the number of points in $CH(S)$, and gave the first algorithm within this time bound. In 1996 Chan [3] gave simpler algorithms for the computation of $CH(S)$ in two and three dimensions. With respect to space complexity, Brönnimann *et al.* [1] showed in 2002 that it is possible to implement Chan's optimal time algorithm with only $O(1)$ additional space to the set of points.

In parallel with the *lower-bound race*, randomized constructions appear to obtain simpler algorithms which are *expected* to run within good time bounds when the input points follow some uniform distribution and/or are processed in random order [6,14].

Many geometric algorithms and structures are based on convex hulls. Therefore, it is not surprising that so much effort has been done in order to lower its computational complexity. In this work we introduce a new technique that, with very little effort from the implementation's point of view, allows many iterative algorithms to be run in parallel. We will concentrate on an incremental randomized construction because of two main reasons: The first is that incremental constructions are usually easy to implement and show very good expected running times. The second reason is that, in an incremental process, many iterations

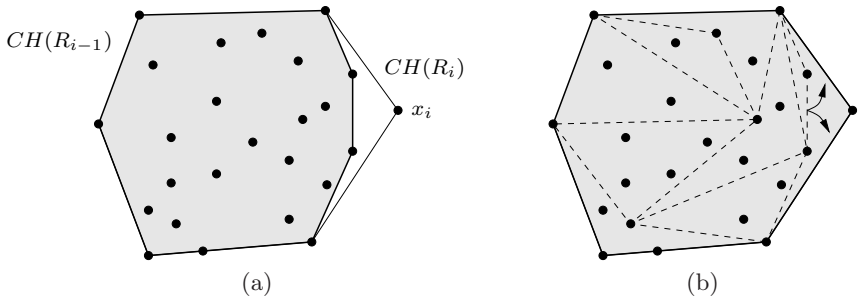


Fig. 1. Clarkson *et al.* algorithm: (a) adding a new point to the convex hull; (b) growth of the convex hull (auxiliary structure shown in dashed lines).

do not change the structure already computed and dependences between processors are relatively rare or, at least, bounded by the number of changes in the structure along the execution.

2.1 Clarkson et al. Algorithm

One of the most efficient and easy to implement randomized incremental algorithms for the construction of the 2-dimensional convex hull, which can be easily extended to higher dimensions, is due to Clarkson, Mehlhorn and Seidel [7]. A brief description of the algorithm follows. More details can be found in [13].

Let S be a set of n points in the plane, let $x_1, x_2, \ldots, x_n$ be a random permutation of the points in S, and call R_i the random subset $\{x_1, x_2, \ldots, x_i\}$. Suppose $CH(R_{i-1})$ is already computed and we want to compute $CH(R_i)$. Point x_i can be inside or outside $CH(R_{i-1})$. If it is inside, obviously, $CH(R_i) = CH(R_{i-1})$. Otherwise, x_i is on the boundary of $CH(R_i)$. All edges in $CH(R_{i-1})$ between the two tangents from x_i to $CH(R_{i-1})$ should be deleted and these two tangents should be added into $CH(R_i)$. See Figure 1(a).

The main idea on Clarkson's *et al.* algorithm is to keep an auxiliary structure that helps finding, in expected $O(\log n)$ time, some edge between the two tangents visible from the new point x_i (see Figure 1(a)) and keeps track of all edges created during the construction of the hull $CH(R_{i-1})$. For each edge in $CH(R_{i-1})$, two pointers are kept for the previous and next edges in the hull. But when an edge should be deleted, these pointers indicate the two new edges in $CH(R_i)$ that caused its deletion. See Figure 1(b).

On each iteration, the algorithm follows the path from the first constructed triangle to the point being inserted and outputs, if it is outside, one edge that is visible from the point. This way, the cost of performing a sequential search for the tangents will be amortized, since all visited edges will be deleted and we can reach the expected $O(n \log n)$ time bound [7].

Think now on a computer with several processors and suppose that we assign each processor one iteration of the algorithm. We expect that only $O(\log n)$ iterations will produce changes in the convex hull. This means that most of

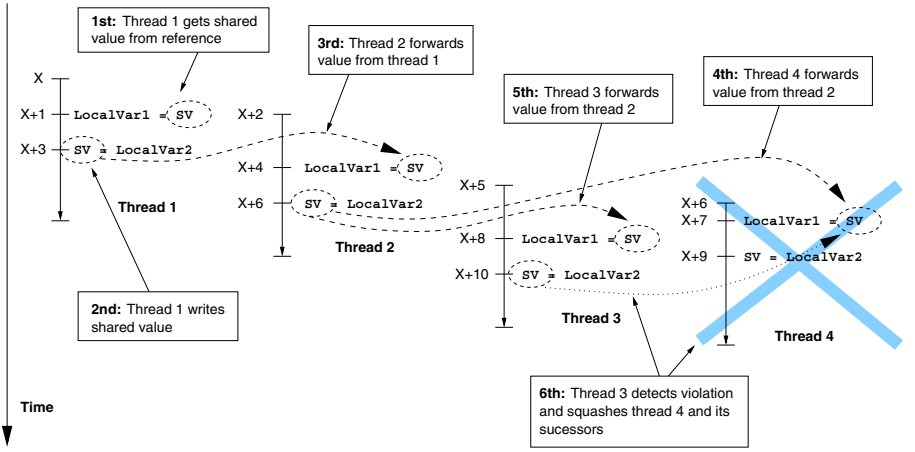


Fig. 2. Speculative parallelization.

the iterations are independent in the sense that they can be run at the same time using the same computed structure. In the next section we introduce a technique called *speculative parallelization*, explaining how this technique can help to speed up the execution of many randomized incremental algorithms sharing this property.

3 Speculative Parallelization

The basic idea under speculative parallelization (also called *thread-level speculation*) [4,12,15] is to assign the execution of different *blocks* of consecutive iterations to different threads, running each one on its own processor. While execution proceeds, a software monitor ensures that no thread consumes an incorrect version of a value that should be calculated by a predecessor, therefore violating sequential semantics. If such a *dependence violation* occur, the monitor stops the parallel execution of the offending threads, discards iterations incorrectly calculated, and restart their execution using the correct values. See Figure 2.

The detection of dependence violations can be done either by hardware or software. Hardware solutions (see e.g., [5,10,16]) rely on additional hardware modules to detect dependences, while software methods [4,12,15] augment the original loop with new instructions that check for violations during the parallel execution. We have presented in [4] a new software-only speculative parallelization engine to automatically execute in parallel sequential loops with few or no dependences among iterations. The main advantage of this solution is that it makes possible to parallelize an iterative application automatically by a compiler, thus obtaining speedups in a parallel machine without the cost of a manual parallelization. To do so, the compiler augments the original code with function calls to perform accesses to the structure shared among threads, and to monitor the parallel execution of the loop.

3.1 Types of Data Dependences

From the parallel execution point of view, in each iteration two different classes of variables can appear. Informally speaking, *private* variables will be those that are always written in each iteration before being used. On the other hand, values stored in *shared* variables are used among different iterations. It is easy to see that if all variables are private, then no dependences can arise and the loop can be executed in parallel. Shared variables may lead to dependence violations only if a value is written in a given iteration and a successor has consumed an outdated value. This is known as the Read-after-Write (RAW) dependence. In this case, the latter iteration and all its successors should be re-executed using the correct values. This is known as a *squash* operation.

To simplify squashes, threads that execute each iteration do not change directly the shared structure: instead, each thread maintains a *version* of the structure. Only if the execution of the iteration succeeds, changes are reflected to the original shared structure, through a *commit* operation. This operation should be done in order for each block of iterations, from the non-speculative thread (that is, the one executing the earliest block) to the most-speculative one. If the execution of the iteration fails, version data is discarded. The next section discusses these operations in more detail.

3.2 Augmenting the Convex-Hull Algorithm for Speculative Execution

Clarkson *et al.* algorithm shown in section 2.1 relies on a structure that holds the edges composing the current convex hull. Whenever a new point is added, the point is checked against the current solution. It is easy to see that this structure should be shared among different iterations. If the point is inside the hull, the current solution is not modified. Otherwise, the new convex hull should be calculated to contain the new edges defined by the point. From the speculative execution point of view, each time a new point modifies the convex hull the parallel execution of subsequent iterations should be restarted, thus degrading performance. Fortunately, as execution proceeds new points are less likely to modify the current solution, and large blocks of iterations can be calculated in parallel without leading to dependence violations. This is why speculative parallelization is a valid technique to speed up the execution of this kind of algorithms.

To compare the performance of the speculative version against the sequential algorithm, we have implemented a Fortran version of Clarkson *et al.* algorithm, augmenting the sequential code manually for speculative parallelization. This task can be performed automatically by a state-of-the art compiler. A complete and detailed description of these operations can be found in [4]. A summary of the changes made in the sequential code follows.

Thread scheduling. For each loop, blocks of consecutive iterations are distributed among different threads.

Speculative loads and stores. As long as each thread maintains its own version copy of the shared structure, all original reads and writes to this structure should be augmented with a procedure call that performs the operation required and checks for possible violations. For example, a read of the shared structure such as `psource = hull(e,source)` should be replaced with the following code:

```
! Calculate linear position of element in shared structure
position = e + NumEdges * (source-1)
! Perform load operation, returning value in "psource"
call specload(position,MyThreadID,psource,hull)
```

Thread commit. After executing a block of iterations, each thread calls a function that checks its state and performs the commit when appropriate.

After augmenting the code for speculative parallelization, we compared its performance with the sequential version under different configurations and with several input sets. Results are shown in the next section.

4 Experimental Results

The experiments performed to measure the execution time of both sequential and parallel versions of the algorithm were done on a Sun Fire 15K symmetric multiprocessor (SMP), equipped with 900MHz UltraSparc-III processors, each with a private 64 KByte 4-way set-associative L1 cache, a private 8 MByte direct-mapped L2 cache, and 1 GByte of shared memory per processor. The system runs SunOS 5.8. The application was compiled with the Forte Developer 7 Fortran 95 compiler using the highest optimization settings for our execution environment: `-O3 -xchip=ultra3 -xarch=v8plusb -cache=64/32/4:8192/64/1`

Times shown in the following sections represent the time spent in the execution of the processing loop of the application. The time needed to read the input set and the time needed to output the convex hull have not been taken into account. The application had exclusive use of the processors during the entire execution and we use wall-clock time in our time measurements.

4.1 Design of the Input: Shape and Size

The number of violations between executions is bounded by the number of points lying outside the convex hull computed up to their insertion. Depending on how quickly the growing convex hull tends to the final one, the number of dependences changes. We have thus designed four different input sets: The first two are sets of 10 and 40 million random points in a square, where we expect violations to lower rather quickly after some iterations; The two others are sets of 10 and 40 million random points in a disk, where the final convex hull is expected to have size $O(\log n)$ and violations will happen more often. We will not analyze degenerate cases like a set of points on a circle, since every iteration is dependent

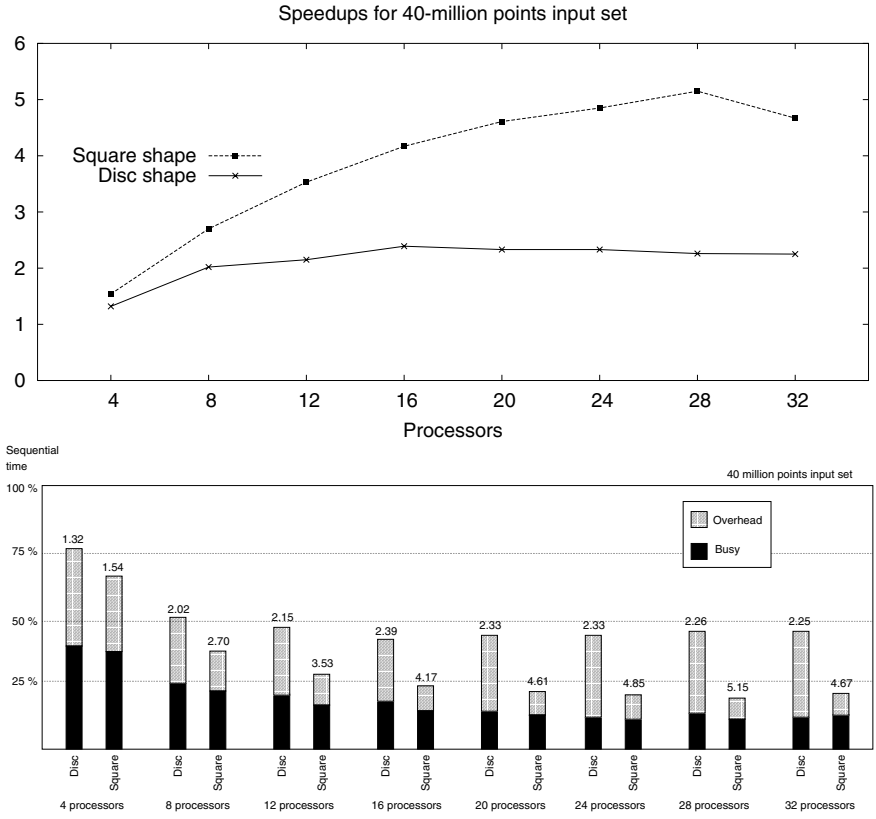


Fig. 3. Speedups and execution breakdown for 40 million points problem size.

on the previous ones and the problem is inherently non-parallel. Smaller input sets were not considered, since their sequential execution time took less than ten seconds in the system under test.

The sets of points have been generated using the random points generator in CGAL 2.4 [2] and have been randomly ordered using its *shuffle* function.

4.2 Overall Speedups

Figure 3 shows the effect of executing the parallel code with the 40 million points problem size for square and disc input sets. Results are normalized with respect to the corresponding sequential execution time. Results are shown for 4 to 32 processors. Execution time breakdowns are divided into "overhead" time (spent in different operations such as synchronization, commit, and loads/stores) and "busy" time that reflects the original loop calculations. Figure 4 shows the effect of executing the parallel code with the 10 million points problem size for the square and disc input sets.

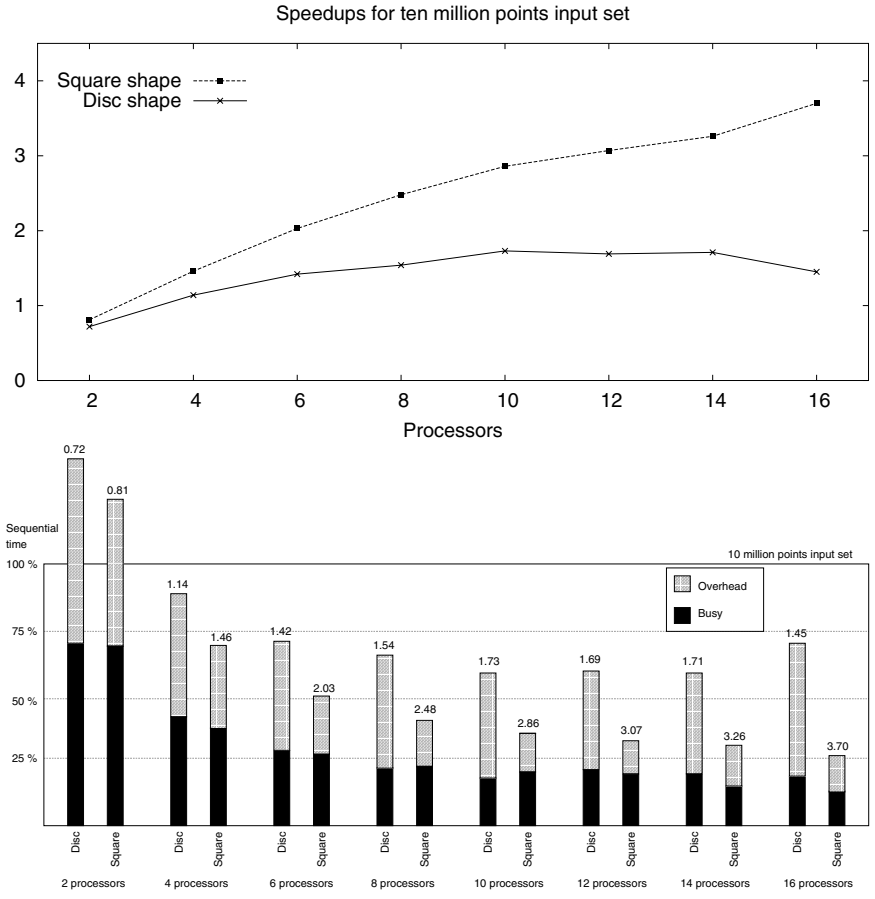


Fig. 4. Speedups and execution breakdown for 10 million points problem size.

From our experiments we can draw the following observations:

- The higher the input size, the higher the speedups obtained, because more and more points can be processed in parallel without modifying the current convex hull.
- The system scales well, allowing better speedups when adding more processors. As can be seen in figure 3, our experiments show a maximum speedup of $5.15x$ with 28 processors for the square input set.
- A significant part of the time is spent in the original calculations. Our experiments shows that the main source of overhead are accesses to the shared structure, in particular load operations.
- As expected, speedups are poorer for the disc input sets, since they have a richer set of edges in the solution, and more memory operations are needed

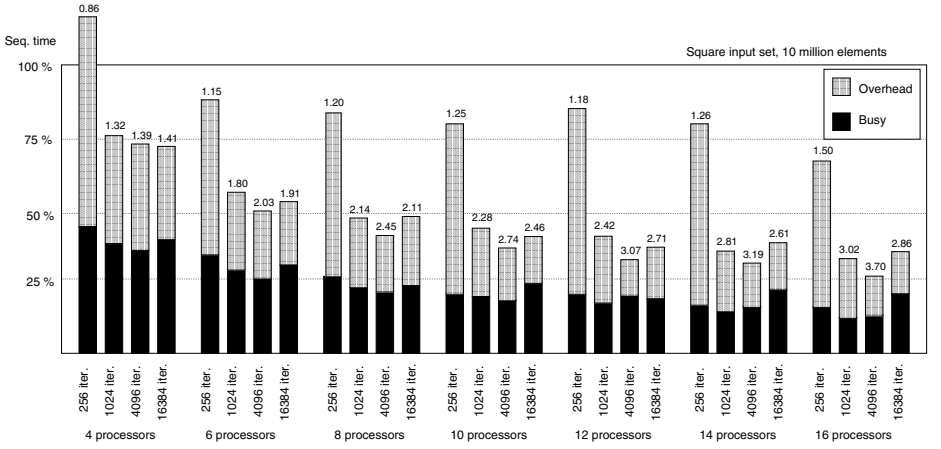


Fig. 5. Execution breakdowns for different block sizes, with a window size equal to the number of processors [4].

to determine whether a given point is inside the current solution. However, we already obtain speedups with as little as four processors.

– Choosing a higher block size does not necessarily lead to better speedups. Optimum block size is a trade-off between having few blocks to execute and having few threads to squash, and also depends on the size of the input set and its shape. Figure 5 shows speedups for different block sizes for one of our input sets. In general, values between 1K and 4K iterations lead to acceptable results for all input sets considered in this work.

5 Conclusions

Parallel implementations of incremental algorithms are hard to develop and require an in-depth understanding of the problem, the language, the compiler and the underlying computer architecture. In this paper we have shown how we can use speculative parallelization techniques to execute automatically in parallel the randomized incremental convex hull algorithm. Choosing an adequate block size, good speedups can be obtained for different workloads with a negligible implementation cost.

Acknowledgments. We would like to thank Pedro Ramos for his helpful comments concerning randomized algorithms.

References

1. H. Brönnimann, J. Iacono, J. Katajainen, P. Morin, J. Morrison, and G. T. Toussaint. In-place planar convex hull algorithms. In *Proc. of the 5th Latin American Symp. on Theor. Informatics (LATIN'02)*, pages 494–507, April 2002.
2. CGAL, Computational Geometry Algorithms Library. `http://www.cgal.org/`.
3. T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete Comput. Geom.*, 16:361–368, 1996.
4. M. Cintra and D. R. Llanos. Toward efficient and robust software speculative parallelization on multiprocessors. In *Proc. of the SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*, pages 13–24, June 2003.
5. M. Cintra, J. F. Martínez, and J. Torrellas. Architectural support for scalable speculative parallelization in shared-memory multiprocessors. In *Proc. of the 27th Intl. Symp. on Computer Architecture (ISCA)*, pages 256–264, June 2000.
6. K. L. Clarkson. Randomized geometric algorithms. In Ding-Zhu Du and Frank Hwang, editors, *Computing in Euclidean Geometry*, volume 4 of *Lect. Notes Series on Computing*, pages 149–194. World Scientific, 2nd edition, 1995.
7. K. L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3(4):185–212, 1993.
8. M. Ghouse and M. Goodrich. Fast randomized parallel methods for planar convex hull construction. *Comput. Geom. Theory Appl.*, 7:219–236, 1997.
9. R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.*, 1:132–133, 1972.
10. L. Hammond, M. Willey, and K. Olukotun. Data speculation support for a chip multiprocessor. In *Proc. of the 8th Intl. Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 58–69, October 1998.
11. D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.*, 15:287–299, 1986.
12. M. Gupta and R. Nim. Techniques for run-time parallelization of loops. *Supercomputing*, November 1998.
13. K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing.* Cambridge University Press, Cambridge, UK, 2000.
14. K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1994.
15. L. Rauchwerger and D. A. Padua. The LRPD test: Speculative run-time parallelization of loops with privatization and reduction parallelization. *IEEE Transactions on Parallel and Distributed Systems*, 10(2):160–180, 1999.
16. G. Sohi, S. Breach, and T. Vijaykumar. Multiscalar processors. In *Proc. of the 22nd Intl. Symp. on Computer Architecture (ISCA)*, pages 414–425, June 1995.

The Employment of Regular Triangulation for Constrained Delaunay Triangulation*

Pavel Maur and Ivana Kolingerová

Department of Computer Science and Engineering,
University of West Bohemia,
Pilsen, Czech Republic,
{maur,kolinger}@kiv.zcu.cz,
http://herakles.zcu.cz/{~maur,~kolinger}

Abstract. We demonstrate a connection between a regular triangulation and a constrained Delaunay triangulation in 2D. We propose an algorithm for edge enforcement in the constrained Delaunay triangulation based on the use of regular triangulation. As far as we know, such a connection has not been presented yet in the literature and there is no algorithm based on this idea, too. This work also serves as a spring-board to higher dimensions.

1 Introduction

A Delaunay triangulation (DT) is one of the fundamental structures in computational geometry. Although it can be defined in an arbitrary dimension, its practical use is mainly in two and three-dimensional space. The nice features of the Delaunay triangulation—mainly optimality properties, which lead to good shapes of Delaunay simplices—has found its applications in FEM computation, object reconstruction, image processing, etc.

A constrained Delaunay triangulation (CDT) arises when arbitrary faces are forced to appear in a Delaunay triangulation. In practice, a CDT is important especially in the cases when the boundary of the triangulated domain has to be kept. Although the CDT loses some of the properties of the DT—e.g. it is not fully Delaunay any more—there are some optimal properties, which still hold.

There are several terms related to the CDT especially because of difficulty of CDT construction in 3D (we mean the terms *conforming DT*, *constrained DT*, *almost DT* or *conforming constrained DT* mentioned in [7,9]). In this paper we use the term constrained DT in the sense that the constraining faces has to be forced without any additional points. More, we restrict ourselves into the 2D.

* This work was supported by the Ministry of Education of the Czech Republic—project MSM 23500005.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 198–206, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Triangulations

2.1 Delaunay Triangulation

Definition 1 (triangulation). *Let us have a set S of points $p = (x_1, x_2)$, $p \in \Re^2$, $card(S) = n$, assuming points in general position (no three points are collinear, no four points are cocircular). A triangulation T of S is a decomposition of the convex hull of S into non-overlapping triangles (the intersection of triangles is either an edge or a vertex or is empty). The vertices of triangles are points of S.*

Definition 2 (Delaunay triangulation). *The triangulation is Delaunay if each triangle satisfies an empty circle property: the circumscribed circle of the triangle does not contain any other point from S in its interior [6]. The same empty circle property holds also for simplices of lower dimensions (edges and points).*

The Delaunay triangulation contains all of the input points. There is a close relation between DT in $\Re^d$ and a convex hull in $\Re^{(d+1)}$. Let us define the set S^+ of points $p^+ = (x_1, x_2, x_1^2 + x_2^2) \in \Re^3$. The points of S^+ lie on the surface of paraboloid. It can be shown that the projection of lower part of conv(S^+) along the z-axis is DT(S) [4].

2.2 Regular Triangulation

Let us define regular triangulations RT(S) following the definitions in [4,5].

Definition 3 (power distance). *Let us consider a point set $S \subset \Re^2 \times \Re$, where each point $p = (x_1, x_2) \in S$ is assigned a weight w_p. A positively weighted point can be interpreted as a circle with the center p and the radius $\sqrt{w_p}$. For each weighted point p we define the power distance of $x \in \Re^2$ to p as $\pi_p(x) = |xp|^2 - w_p$, where $|xp|$ is the Euclidean distance from x to p, see Fig. 1.*

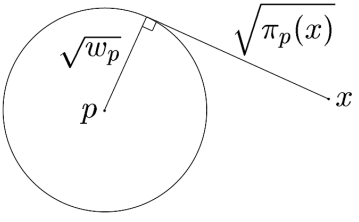


Fig. 1. The power distance $\pi_p(x)$ from x to p; w_p is the weight of p.

Definition 4 (orthogonality). *Two weighted points p and z are called orthogonal, if* $|pz|^2 = w_p + w_z$. *It means* $\pi_p(z) = w_z$ *and* $\pi_z(p) = w_p$. *Let three weighted points define a triangle A. There is a unique weighted point z that is orthogonal to all weighted points of triangle A. This point is called the orthogonal center of A. If the weights of all points of A are equal to zero then the circle with the center z and radius* $\sqrt{w_z}$ *is the circumcircle of A. For all* $p \in A, \pi_z(p) = w_p$ *by definition.*

Definition 5 (regular triangulation). *The triangle A is (globally) regular if* $\pi_z(q) > w_q$ *for all* $q \in S \setminus A$. *The set of regular triangles defines the regular triangulation of S.*

It can be shown that dual of RT(S) is a power diagram of S. If the weights of all points are zero, the RT becomes DT and the power diagram becomes the Voronoi diagram.

RT in $\Re^d$ maintains the relation to the convex hull in $\Re^{(d+1)}$. Let us define the lifted point set $S^+ \subset \Re^3$ again, where the points $p^+ \in S^+$ are defined as $p^+ = (x_1, x_2, x_1^2 + x_2^2 - w_p)$. As the weight is subtracted from the z-coordinate, not all points p^+ lie on the paraboloid (recall that $w_p \in \Re$, thus p^+ can be placed below as well as above the paraboloid). If p^+ is not incident on any of triangles of the lower convex hull of S^+ then p^+ is called a *redundant point* and is not present in RT(S).

2.3 Constrained Delaunay Triangulation

A constrained Delaunay triangulation arises from DT, but brings several changes [2]. First, the CDT is not Delaunay triangulation, because there are simplices that are not Delaunay. Second the CDT must include prescribed faces, so-called *constraints* or *constraining faces*. Let us follow the definition of CDT in 2D from [7].

Definition 6 (constrained Delaunay triangulation). *The input of CDT is a planar straight line graph (PSLG) X, which is a set of vertices and segments (constraining edges). A CDT contains only vertices from X and every segment of X is a single edge of CDT. The Delaunay (or empty circle) condition is changed as follows: every simplex must be either a segment prescribed in X or be constrained Delaunay. A simplex is constrained Delaunay if it has a circumcircle that encloses no vertex of X that is visible from any other point in the relative interior (i.e. excluding the boundary) of the simplex. More, the relative interior of the simplex does not intersect any segment. Visibility is occluded only by segments of X.*

As well as an ordinary DT has its dual in the Voronoi diagram, a CDT has also its own dual in the extended Voronoi diagram [3]. CDT in dimensions higher than two is more complicated, because there are polytopes that cannot be triangulated at all without additional vertices. It is no wonder that the generalization of the CDT into higher dimension is quite new [7]. The existence of 2D CDT is proved for an arbitrary input PSLG, for higher dimensional CDT there is a condition, which guarantees its existence [7].

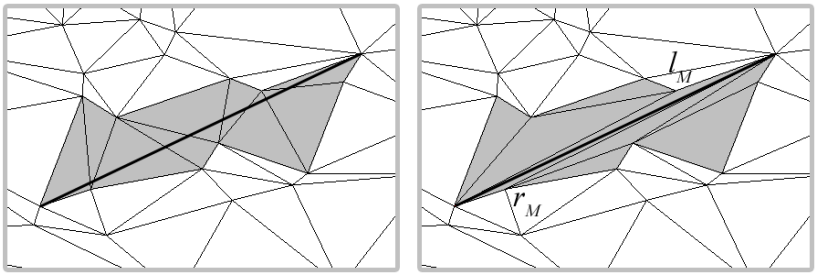


Fig. 2. The area affected by the constraining edge in the Delaunay triangulation (left) and the constrained Delaunay triangulation after insertion of the constraining edge (right). The triangulation does not change outside the affected area (marked with grey color).

3 Regular Triangulation Can Serve for Constrained Delaunay Triangulation

There are two ways to construct a CDT. First, use the algorithm that builds a CDT directly [2,8] or second, start with the DT, which is then processed to contain constraining edges [11,1]. In this section we show that the RT can be employed in the second class of algorithms.

There is no way how to describe the whole CDT as the RT, because the CDT does not form a convex shape when lifted one dimension higher. Indeed, the convex shape is formed by an ordinary DT, where the points are lifted onto paraboloid. But when a constraining edge is forced into the DT, such a lifted edge makes a local non-convexity in the paraboloid. The part of DT, which is affected by the insertion of the edge, consists of the triangles intersected by the given edge [1], see Fig. 2. If we consider this area as a part separated from the rest of the DT we are able to describe a CDT for it as a regular triangulation. To describe a regular triangulation means to assign the weights to the vertices, as their positions are already given.

Let us denote vertices of the constrained edge e as c_1 and c_2, points of affected area left to the e as l_j and right to the e as r_k following the definition of mutual position of points in [10]. Let us denote all the lifted points by + sign, e.g. p^+ is p lifted. Let us have two paraboloids P_l and P_r with apices p_l and p_r. Both paraboloids have the axes in the positive z-direction and the apices in the x-y plane. Each of them is given by the equation $z = x^2 + y^2 + mx + ny + p$.

In the CDT the constraining edge divides the affected area into two independent Delaunay triangulations, as they are invisible to each other from the definition. Each particular DT can be mapped onto its own paraboloid: all points l_j^+ to P_l and all points r_k^+ to P_r. More, the points c_1^+ and c_2^+ must be mapped onto P_l as well as P_r as they are part of both—left and right—DT. The projection of $P_l \cap P_r$ in the x-y plane is the constraining edge e.

Let us denote points that form triangles with the edge e in the left and right part of CDT as l_M and r_M, respectively (see Fig. 2). From the definition of reg-

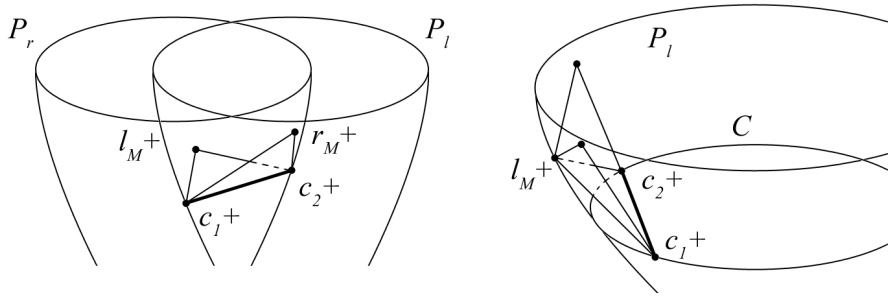


Fig. 3. Left: paraboloids P_r and P_l intersecting in edge $c_1^+c_2^+$. Right: the situation on the paraboloid P_l in more detail.

ular triangulation it must pay that all the points r_k^+ lie in the negative halfspace given by the plane $c_1^+c_2^+l_M^+$ and all the points l_j^+ lie in the negative halfspace given by the plane $c_2^+c_1^+r_M^+$ to keep the convexity of the lifted polyhedron. For the illustration see Fig. 3.

Theorem 1 (local equivalence of CDT(S) and RT(S)). *Given a DT(S) in 2D and a constraining edge e, it is possible to describe the CDT of the area affected by e as the regular triangulation.*

Proof (local equivalence of CDT(S) and RT(S)). Without loss of generality let us assume the points c_1^+ and c_2^+ lie in a horizontal plane α. If the point l_M^+ lied in α, the intersection of P_l and α would appear as the circle C. In the projection into the x-y plane there are no other points l_j inside C and the center c of C is a projection of apex p_l of P_l (in fact it is p_l because—from the definition—the z-coordinate of p_l is zero). If l_M^+ lies above the plane, the projection of C becomes completely free of l_j. The same holds for the set of points r_k. Thus, we have found a regular triangulation, which consists of two independent Delaunay triangulations separated by the constraining edge e. The lifted DTs lie on two paraboloids P_l and P_r. If all l_j^+ and r_k^+ lie above the plane α, the convexity and thus the existence of lifted RT is ensured.

4 Algorithms

Let us say now several words about the algorithms for constructing DT. Many different approaches for DT in 2D were invented up to now. Among others there is a class of algorithms called *flip-algorithms*, which are based on diagonal swap of adjacent triangles forming strictly convex quadrilateral. Only one diagonal can satisfy Delaunay criterion—this is clear if we realize that a strictly convex quadrilateral is nothing else than a 2D projection of a tetrahedron, whose vertices are lifted on a paraboloid. Only the lower part of the tetrahedron is valid for Delaunay triangulation, because it is convex (see Fig. 4).

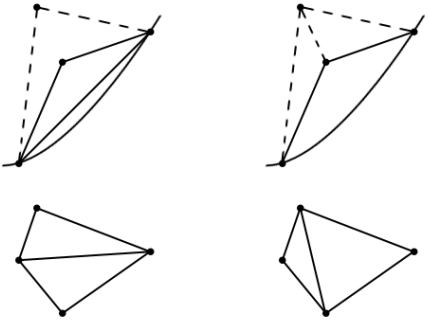


Fig. 4. To flip the diagonal in 2D means to select one of two halves of lifted tetrahedron. The dashed lines are inside the paraboloid.

It is proved (Lawson) that in 2D it is possible to start with an arbitrary triangulation and after a finite number of flips the DT appears. On the other hand it is proved that such approach does not work in higher dimensions. In 3D and higher dimensions the flipping procedure has to be joined with incremental construction to work properly (Joe) otherwise the flipping can get stuck. In fact, the flipping procedure processes locally non-optimal faces towards the optimal ones. In higher dimensions the situation can appear where non-optimal faces are not flippable, which stops the flipping before the DT is reached. The time optimal complexity of DT algorithms in 2D is $O(n \log n)$, in higher dimensions depends on the number of output simplices and is $O(n^{\lceil d/2 \rceil})$ in the worst case.

Edelsbrunner and Shah [4] proved that for RT even in 2D the flipping must also be joined with the incremental construction. Otherwise there may appear non-regular faces, which are not swappable. For the general construction of RT so called generalized flips are needed, which deal with introducing and disappearing of the redundant points. This technique is not used in our method and thus is not described here. We can refer readers to [4]. Flipping based incremental algorithms work for arbitrary dimension with the expected time complexity $O(n \log n + n^{\lceil d/2 \rceil})$ [4].

Now let us concentrate on algorithms for CDT, especially on the second class mentioned before. Essentially, there are two methods that are able to force an edge into a DT:

- use of flips,
- retriangulation of the affected area.

In the first approach, the triangles intersected by the intended constrained edge are flipped so long until the edge appears in the triangulation [11]. The second approach removes all the triangles intersected by the constrained edge from the triangulation, establishes the edge and retriangulates the holes on both side of the edge [1]. The time optimal complexity of CDT algorithms in 2D is the same as for Delaunay triangulation.

5 Our Approach to Edge Enforcement

Our algorithm, which utilizes the facts mentioned before, is based on the flipping procedure. The advantage of our algorithm is that there is no need to check explicitly the existence of constraining edge during the flipping. The constraining edge appears automatically after the regular triangulation is constructed. Applied RT is constructed without redundant points, which follows from its definition. The whole algorithm is as follows:

input:
the set of points $S = (p_0, p_1, \ldots, p_{n-1}) \in \Re^2$;
the set of constraining non-intersecting edges $E = (e_0, e_1, \ldots, e_{m-1})$,
$e_j = (p_u, p_v)$, $u \neq v$;
output:
the constrained Delaunay triangulation of S and E;

construct DT(S);
for each constraining edge do
{
  find and separate the affected area A;
  find apices of paraboloids P_l and P_r;
  set the w_{p_i} for $p_i \in A$ according to the particular paraboloid;
  construct RT of the affected area by flipping;
  reset the w_{p_i};
  fix the constraining edge;
{

Proof (convergence of the algorithm). To prove the convergence of the whole algorithm only the part using the flipping procedure has to be proved to converge, because we do not use the incremental construction of regular triangulation. In [4] such a regular triangulation in 2D is presented, whose all non-regular edges are not swappable—there is no way how to regularize this triangulation with the help of flipping, see in Fig 5. The reason for this is clear—the lifted triangulation

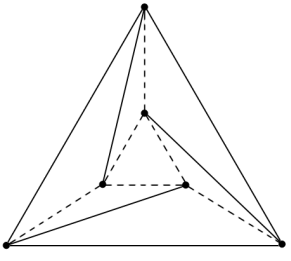


Fig. 5. The non-regular non-flippable triangulation [4], where the solid edges are locally regular, while the dashed edges are non-regular and non-flippable.

creates the Schönhardt polyhedron, which is the most famous untetrahedralizable polyhedron. As mentioned before, the diagonal flip in 2D means the selection of the upper or lower part of tetrahedron, whose points are lifted into 3D. If the lifted triangulation forms a polyhedron, which is not tetrahedralizable, the flipping procedure is not able to converge towards the regular triangulation.

The existence of 2D CDT is proved [7] for any input data. It implies there always exists the tetrahedralization of the polyhedron, whose boundary is created by union of the lifted DT and lifted CDT of affected area. Thus, it is always possible to force the edge by flipping.

6 Conclusion

We propose a new point of view of the constrained Delaunay triangulation, which can be locally described as a regular triangulation. We give a description, how to set the weights of the affected points as well as the whole algorithm for the constraining edge forcing, which is based on a flipping procedure.

Although we started our research in 2D, the main goal of our future work is to extend this method into the 3D or higher dimensions to provide a simple algorithm for the CDT in an arbitrary dimension.

Acknowledgement. The authors would like to thank to Václav Skala from University of West Bohemia in Pilsen, Czech Republic, for his material and moral support, to Andrej Ferko from Comenius University in Bratislava and our colleagues Josef Kohout and Petr Vaněček for their comments. And also to Jonathan R. Shewchuk from University of California at Berkeley for his unbelievable working assignment and results, which are always great inspiration for us.

References

1. Anglada, M. V.: An improved incremental algorithm for constructing restricted Delaunay triangulations. Comput. & Graphics, Vol. 21, No. 2, 1997, pp. 215–223.
2. Chew, L. P.: Constrained Delaunay Triangulations. Proceedings of 3rd Annual Symposium on Computational Geometry, ACM, 1987.
3. Edelsbrunner, H.: Triangulations and meshes in computational geometry. Acia Numerica, 2000, pp. 1–81.
4. Edelsbrunner, H., Shah, N. R.: Incremental Topological Flipping Works for Regular Triangulations. Proceedings of the 8th Annual Computational Geometry, ACM, 1992, pp. 43–52.
5. Facello, M. A.: Implementation of a randomized algorithm for Delaunay and regular triangulations in three dimensions. Computer Aided Geometric Design 12, 1995, pp. 349–370.
6. Preparata, F. P. and Shamos, M. I.: Computational Geometry, Springer-Verlag, 1985.
7. Shewchuk, J. R.: A Condition Guaranteeing the Existence of Higher-Dimensional Constrained Delaunay Triangulations. Proceedings of the Fourteenth Annual Symposium on Computational Geometry, ACM, 1998, pp 76–85.

8. Shewchuk, J. R.: Sweep Algorithms for Constructing Higher-Dimensional Constrained Delaunay Triangulations. Proceedings of the Sixteenth Annual Symposium on Computational Geometry ACM, 2000, pp. 350–359.
9. Shewchuk, J. R.: Constrained Delaunay Tetrahedralizations and Provably Good Boundary Recovery. To appear in the 11th International Meshing Roundtable, 2002.
10. Shewchuk, J. R.: Robust Adaptive Floating-Point Geometric Predicates. Proceedings of the Twelfth Annual Symposium on Computational Geometry, ACM, 1996.
11. Sloan, S. W.: A Fast Algorithm for Generating Constrained Delaunay Triangulations. Computers & Structures, Vol. 47, No. 3, 1993, pp. 441–450.

The Anchored Voronoi Diagram

Jose Miguel Díaz-Báñez[1], Francisco Gómez[2], and Immaculada Ventura[3]

[1] Universidad de Sevilla, SPAIN,
dbanez@us.es
[2] Universidad politécnica de Madrid, SPAIN,
fmartin@eui.upm.es
[3] Universidad de Huelva, SPAIN
inmaculada.ventura@dmat.uhu.es

Abstract. Given a set S of n points in the plane and a fixed point o, we introduce the Voronoi diagram of S anchored at o. It will be defined as an abstract Voronoi diagram that uses as bisectors the following curves. For each pair of points p, q in S, the bisecting curve between p and q is the locus of points x in the plane such that the line segment $\overline{ox}$ is equidistant to both p and q. We show that those bisectors have nice properties and, therefore, this new structure can be computed in $O(n \log n)$ time and $O(n)$ space both for nearest-site and furthest-site versions. Finally, we show how to use these structures for solving several optimization problems.

1 Introduction

Given a set of n sites in a continuous space, the subdivision of the space into regions, one per site, according to some influence criterion is a central topic in Computational Geometry and such divisions have been applied to many fields of science. The standard name for this geometric structure is due to Voronoi, who proposed the first formalization. Originally, this structure was used for characterizing regions of proximity for the sites. Since then, many extensions and generalizations have been proposed (see the surveys [1,6,9]). Also, other general approachs have been introduced [5,7] where the concepts of site or distance functions are not explicitly used. In this paper, we introduce an abstract Voronoi diagram in the sense of [7], *the anchored Voronoi diagram*. In section 2, we formally define this structure and give some structural properties. In Section 3 we show how to compute it. Section 4 is devoted to describing the properties and the computation of the furthest-site anchored Voronoi diagram. We show in Section 5 how to apply this structure for solving some facility location problems. Those problems consist of finding the anchored bridge that connects a point with a curve so that the distance from the bridge to a given point set is maximized or minimized.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 207–216, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 The Anchored Voronoi Diagram

2.1 Definition and Properties

We begin by introducing some notation. Given a set S of n points in the plane, the Euclidean distance between two points p and q will be denoted by $d(p,q)$ and the Euclidean distance between a point p and the origin will be denoted by $\|p\|$. We define an *anchored* segment as a line segment when the initial point is fixed. Without loss of generality, we will consider the anchor to be the origin. Throughout this paper, we suppose that $o \notin S$. Finally, the distance between a point $p \in S$ and an anchored segment connecting o with a point $x \in \mathbb{R}$ will be defined as $d(p, \overline{ox}) := \min\{d(p,q) : q \in \overline{ox}\}$.

The structure to be constructed intrinsically depends on the above distance. In order to make easier later descriptions, we introduce some geometric tools. First, we show a geometric rule for computing the distance between a point and an anchored segment. Given a point p of S, let C_p be the circle of radius $d(o,p)/2$ centered at the midpoint of the segment $\overline{op}$, let l be the line through o perpendicular to $\overrightarrow{op}$ and denote by H_p the halfplane of the line l in which p does not lie (as illustrated in Figure 1). Then, in order to calculate the distance between p and the segment $\overline{ox}$, ($x \in \mathbb{R}^2$), we proceed as follows: If x is inside C_p, then $d(p, \overline{ox}) = d(p,x)$. If $x \in H_p$, then $d(p, \overline{ox}) = d(p,o)$. Otherwise, $d(p, \overline{ox}) = d(p,r)$ where r is the intersection point between $\overline{ox}$ and C_p. The following locus will be useful in the rest of the paper. Given an anchored segment $\overline{ox}$ and $\varepsilon \geq 0$, the locus of points equidistant from $\overline{ox}$ at distance ε is called an anchored hippodrome centered at $\overline{ox}$ of radius ε.

As pointed out above, in this paper we introduce a new Voronoi diagram by means of a set of bisecting curves. For any two different points p, q in S, a *bisecting curve* $L(p,q)$ is defined as the locus of points x in the plane such that the line segment $\overline{ox}$ is equidistant to both p and q, that is, $L(p,q) = \{x \in \mathbb{R}^2 : d(p, \overline{ox}) = d(q, \overline{ox})\}$.

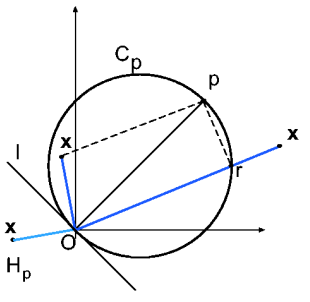


Fig. 1. Geometric rule for computing the distance.

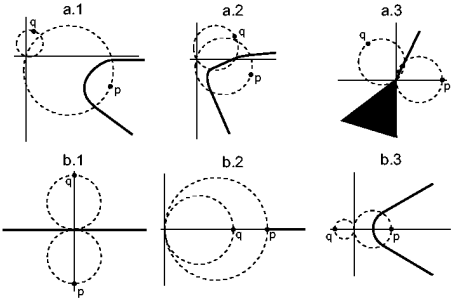


Fig. 2. The locus $L(p,q)$.

By using the above geometric rule we can generate $L(p,q)$. An exhaustive study of the properties and the shape of $L(p,q)$ have been carried out in [2]. $L(p,q)$ dissects the plane into two open (unbounded) domains $D(p,q)$ and $D(q,p)$ having both of them $L(p,q)$ as complete separating boundary. More precisely, $D(p,q) = \{x \in \mathbb{R}^2 : d(p,\overline{ox}) < d(q,\overline{ox})\}$ and $D(q,p) = \{x \in \mathbb{R}^2 : d(q,\overline{ox}) < d(p,\overline{ox})\}$. In [2], the authors give a complete proof of the fact that, if $d(o,p) < d(o,q)$, then $D(q,p)$ is a convex region. Six cases have been obtained depending on whether p, q and o are in line or not and $\|p\| \neq \|q\|$ or not.

A generic curve $L(p,q)$ is a continuous curve consisting of a half-line, an arc of a curve of degree four, an arc of a circle (which may or may not exist) and finally another half-line. In Figure 2, all types of bisecting curves are shown. Note that in the case in which the segments $\overline{op}$ and $\overline{oq}$ lie in different lines and $d(o,p) = d(o,q)$, (case (a.3) in Figure 2) $L(p,q)$ includes a region. In fact, the bisecting curve bifurcates at the origin and the entire region bounded by the two branches of the curve (two haflines anchored at the origin) is equidistant from the two points p and q. An other special case is (b.2) in which the region $D(p,q)$ has an empty interior (as subset of $I\!R^2$).

Cases (a.3) and (b.2) are indeed degenerate in the sense that the bisecting curves no longer are curves themselves but regions and domains can be empty. Those degenerate cases can be removed by using the linear scheme of Canny and Emiris [3] seems to be the most appropriate inasmuch as it is simple and easily suitable to our problem. Note that the degenerate cases in our problem are quite simple to detect (it suffices an evaluation of a low-degree polynomial): two points have the same distance to the origin and two points and the origin are collinear. That perturbation method does not increase the time complexity of our algorithms. Hereafter we suppose that the set S does not contain points in degenerate position and then we deal with bisecting curves of cases (a.1), (a.2), (b.1) and (b.3).

Our aim is to define an abstract diagram that fits the framework of [7]. In fact, the family $\mathcal{D} = \{D(p,q), p \neq q\}$ is *dominance system* over S. Thus, we can define the abstract nearest site Voronoi diagram associated to the system of curves $L(p,q)$ as follows:

Definition 1. *For a point $p \in S$, the Anchored Voronoi Region $AVR(p,S)$ is defined as the intersection of the domains $D(p,q)$, where $q \in S \setminus \{p\}$, $AVR(p,S) = \cap_{q \in S, p \neq q} D(p,q)$. The Anchored Voronoi Diagram $AVD(S)$ with respect to the bisecting curves $L(p,q)$ is defined as the union of all boundaries of at least two Voronoi region have in common, $AV(S) = \cup_{p \in S} \delta(AVR(p,S))$.*

We assume an ordering of the points of S and that every portion of the bisecting curve $L(p,q)$ is put in the region of $\min\{p,q\}$. The common boundary of two anchored Voronoi regions is called an anchored Voronoi edge and the common boundary of three anchored Voronoi regions is called an anchored Voronoi vertex, as usual. In Figure 3 an example of an anchored Voronoi diagram is shown for a set of four points. Observe that it does not contain vertices and the nearest point to the origin is neighbor to the rest of the points.

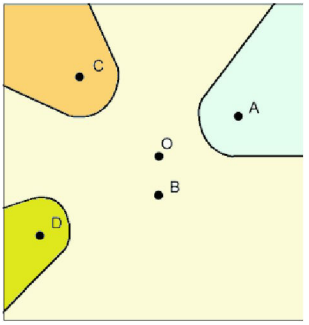


Fig. 3. $AVD(S)$ for four points.

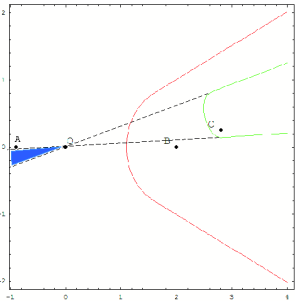


Fig. 4. $AVR(B)$ is non star-shaped.

2.2 Topological Properties

In the following we investigate some topological properties of the diagram AVD(S) which are useful in later sections. We will call the dual graph of $AVD(S)$ *the anchored graph*, $AG(S)$ (whose nodes are the points of S and whose edges connects points with adjacent anchored Voronoi regions). Let observe that $AG(S)$ may does not generate a triangulation of the space. In Figure 3, $AG(S)$ is a tree rooted at point b.

It is well know that for a concrete Voronoi diagram respect to a nice metric [7] the Voronoi cell of a point p is always a star-shaped region whose kernel contains p. In contrast, here there exists Voronoi cells of the AVD(S) which are not star-shaped. Figure 4 shows that the region that corresponds to the site b is not star-shaped. In fact, the visibility region (kernel) $AVR(b)$ (the shadow region) is outside the region.

In the following we study the shape of edges and vertices of the anchored Voronoi diagram. An edge can be composed into pieces which are either half-lines, or line segments, or arcs of a curve of degree four or arcs of a circle.

Lemma 1. *Given three points p, q, r of S, $L(p,q) \cap L(p,r)$ is a connected set.*

Proof. Let us assume for the sake of a contradiction that $L(p,q) \cap L(p,r)$ has more than one connected component (p, q, r are all distinct). Let a and b two points in two different connected components. From the definition of bisecting curve, the following facts are true: $d(p, \overline{oa}) = d(q, \overline{oa}) = d(r, \overline{oa})$ and $d(p, \overline{ob}) = d(q, \overline{ob}) = d(r, \overline{ob})$; points p, q, r belongs to two hippodromes H_1, H_2, centered at $\overline{oa}$ and $\overline{ob}$, respectively; p, q, r belong to $H_1 \cap H_2$. We consider two cases:

1. The origin and a and b are collinear. Let us suppose the situation depicted in Figure 5 (a), that is, $int(H_1) \subset int(H_2)$. Then, b could be continuously moved along line segment $\overline{ab}$ until reaching point a. This is a contradiction for it would imply a and b are in the same connected component of $L(p,q) \cap$

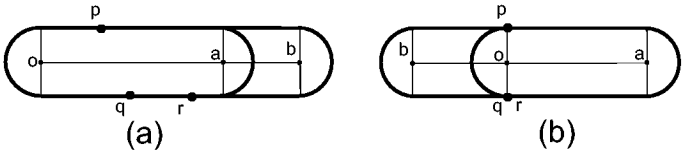


Fig. 5. The origin, a and b are collinear.

$L(p,r)$. Assume then that $int(H_1) \not\subset int(H_2)$ (see Figure 5 (b)). Then, $H_1 \cap H_2$ is reduced to two points and again there is contradiction because we are considering three different points p, q, r.

2. The origin and a and b are not collinear. Consider first the case in which the radii of both hippodromes are the same. In this case, there must be two points in $\{p, q, r\}$ on the arc of circle centered at the origin. Note that the intersection of both hippodromes is an isolated point and arc of a circle. The two points on the arc of circle are at the same distance from the origin. However, such degenerate case cannot occur since it was removed with the method of Emiris and Cannis. Finally, let have H_1, H_2 different radius (for example, radius of H_1 less than that of H_2; the other case is similar). In this case, a little thought reveals that $H_1 \cap H_2$ consists of two points and we obtain a contradiction, since $H_1 \cap H_2$ must contain p, q, r.

Lemma 2. *A vertex of $AVD(S)$ (defined by at least three points) can be defined by a point, a half-line or a curve composed by an arc of circle plus a half-line.*

Proof. As a consequence of Lemma 1, the intersection of two bisecting curves $L(p,q)$, $L(p,r)$ must be either a point or connected subset of a bisecting curve. This means that if such intersection is not a single point, then it must be a half-line, and arc of a circle plus a half-line. We can discard the other possibilities by using the equidistance condition.

3 Computing the Diagram

We now address the construction of the AVD(S). For computing this structure, the divide & conquer approach given in [7] can be used. In fact, we will prove that our set of bisecting curves fulfills the good properties claimed in [7].

Definition 2. *The system $\mathcal{L} = \{L(p,q) : p, q \in S, p \neq q\}$ is called admissible iff for each subset S' of S of size at least 3 the following conditions are fulfilled: (a) The intersection of two bisecting curves only consists of finitely many components; (b) the Voronoi regions are path-connected and (c) Each point of the plane lies in a Voronoi region or on the Voronoi diagram.*

Condition (a) and (c) of above definition immediately holds. The condition (b) is more complicated to prove. Let us give some notation and make some observations in order to simplify our explanation.

Given a point x in the plane and a site $p \in S$, we denote by x' the point in $\overline{ox}$ where the distance from p is attained, then $d(p, \overline{ox}) = d(p, x')$. We also denote by $\mathcal{H}_{x,p}$ the hippodrome centered at $\overline{ox}$ and radius $\varepsilon = d(p, x')$.

Remark: If $x \in AVR(p)$, then the hippodrome $\mathcal{H}_{x,p}$ does not contain any other site in its interior.

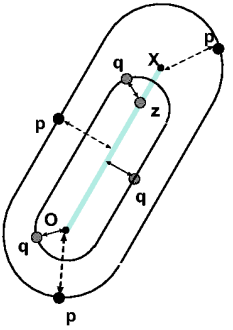


Fig. 6. $\overline{xx'} \subset VR(p)$.

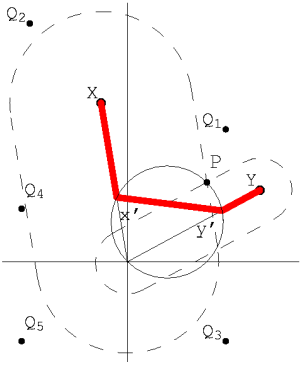


Fig. 7. A polygonal path into $AVR(p)$.

Lemma 3. *Given two points $x, y \in AVR(p)$, the polygonal path with vertices $\{x, x', y', y\}$ is completely contained into $AVR(p)$.*

Proof. We show that the cell $AVR(p)$ contains every line segment of the proposed path of Figure 7. First we prove that $\overline{xx'} \subset AVR(p)$. Let z be a point of the line segment $\overline{xx'}$. Suppose that z is associated to other site $q \in S$, $z \in AVR(q)$. Then the radius of the hippodrome $\mathcal{H}_{z,q}$ is lower than the radius of $\mathcal{H}_{x,p}$ (refer to Figure 6). As consequence, $q \in int(\mathcal{H}_{x,p})$ in contradiction with the above Remark. In a similar way we can prove that $\overline{yy'} \subset AVR(p)$.

The proof of $\overline{x'y'} \subset AVR(p)$ requires more details. A first observation is that the points x' and y' lie in the circle of diameter $d(o, p)$ and passing through o and p, C_p. But this implies that the segment $\overline{x'y'}$ lies in the interior or on the circle C_p that, in turn, implies that for every $z \in \overline{x'y'}$, $d(p, \overline{oz}) = d(p, z)$. On the contrary, given other site $q \in S$, the distance between q and $\overline{oz}$ can be attained either at the end point z, at the origin o or either at the line containing the segment $\overline{oz}$ (to be denoted by R_z). Consider now the partition of the line segment $\overline{x'y'}$ generated by the changes of the distance function $d(q, \overline{oz})$ (as showed in the Figure 8). Note that this function is continuous for all the points $z \in \overline{x'y'}$.

We next show that for every $z \in \overline{x'y'}$, $d(p, \overline{oz}) < d(q, \overline{oz})$ holds. Our argument depends on each element of such partition. We have three different cases:

- Case 1.- Suppose that $d(q, \overline{oz}) = d(q, z), \forall z \in \overline{l_{i-1}l_i} \subset \overline{x'y'}$. In this situation, the bisector $B(p, q)$ (perpendicular line to $\overline{pq}$ passing through the midpoint of $\overline{pq}$) dissects the plane into two halfplanes $H(p, q)$ (that contains to p), and

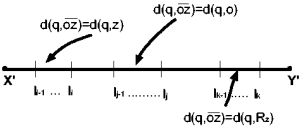


Fig. 8. Partition of the line segment $\overline{x'y'}$.

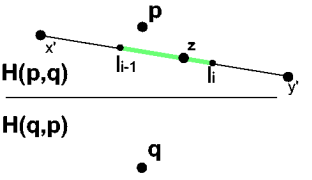


Fig. 9. Proof of Case 1.

$\overline{H(q,p)}$ (containing to the site q)(see Figure 9). Since we have proved that $\overline{xx'}, \overline{yy'} \subset VR(p)$, then $x', y' \in H(p,q)$. This implies that $\overline{x'y'} \subset H(p,q)$ and $\overline{l_{i-1}l_i} \subset \overline{x'y'} \subset H(p,q)$. Finally, $d(p,\overline{oz}) < d(q,\overline{oz})$ and $z \in VR(p)$.

- Case 2.- Suppose $d(q,\overline{oz}) = d(q,o)\ \ \forall z \in \overline{l_{j-1}l_j} \subset \overline{x'y'}$. Consider the triangle of vertices x', p, y' as in Figure 7. Given a point $z \in \overline{l_{j-1}l_j} \subset \overline{x'y'}$, it is easy to obtain that $d(p,\overline{oz}) \leq \max\{d(p,\overline{ox'}), d(p,\overline{oy'})\}$. We now show that $d(p,\overline{ox'}) < d(q,o)$ (in a symmetric way that $d(p,\overline{oy'}) < d(q,o)$). If $d(p,\overline{ox'}) \geq d(q,o)$, then we have that $q \in int(\mathcal{H}_{x',p})$, contradicting the fact that $x' \in AVR(p)$. Therefore, $d(p,\overline{oz}) < d(q,\overline{oz})$ and $z \in AVR(p)$.
- Case 3.- Let $\overline{l_{k-1}, l_k}$ be a subinterval of the partition $\overline{x'y'}$ such that $d(q,\overline{oz}) = d(q,R_z)\ \forall z \in \overline{l_{k-1}, l_k}$. Due the continuity of the distance function and the above proofs, the endpoints l_{k-1} and l_k lie in the cell $VR(p)$, in other words, $d(p,l_{k-1}) < d(q,l_{k-1})$ and $d(p,l_k) < d(q,l_k)$. On the other hand, the distance between a point and an anchored line is a monotone function. Thus $d(q,R_z)$ is a monotone function. Finally, it is easy to prove that $d(p,\overline{oz}) = d(p,z)$ is a convex function for $z \in \overline{x'y'}$. Plugging all together we can see that the graphs of functions $d(p,\overline{oz})$ and $d(q,\overline{oz})$ do not intersect for $z \in \overline{l_{k-1}, l_k}$. As a consequence, $d(p,z) < d(q,R_z) = d(q,\overline{oz})$ and the claim follows.

The above results establish that our system of bisecting curves $\mathcal{L}$ is an admissible system. This allows us to apply the algorithm of [7] and we have the following theorem.

Theorem 1. *The Anchored Voronoi Diagram of a set of point S in the plane can be constructed in $O(n \log n)$ time and $O(n)$ space.*

4 The Furthest-Site Anchored Voronoi Diagram

In this section we address the construction of the furthest-site Voronoi diagram with respect to the system of bisecting curves $L(p,q)$. For this purpose we use the framework of Melhorn et al. [8], which follows Klein's approach for Voronoi diagrams. In [8] is shown that the furthest site Voronoi can also be defined by means of a dominance system.

Definition 3. *Let $\mathcal{L}$ be the system of locus $L(p,q)$ and let $\mathcal{L}^*$ be the "dual" of $\mathcal{L}$, in which both the dominance relations and the ordering of points are reversed.*

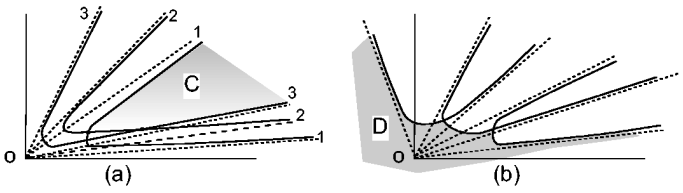


Fig. 10. Intersection of convex and non-convex regions.

$AVR^*(p,S) = \cap_{q\in S, p\neq q} D(q,p) = \{x \in I\!R^2 : d(p,\overline{ox}) > d(q,\overline{ox}),$ *for all* $q \in S \setminus \{p\}\}$. $AV^*(S) = \cup_{p\in S}\delta(AVR^*(p,S))$.

We call $AVR^*(p,S)$ (hereafter denoted by $AVR^*(p)$) the furthest site anchored Voronoi region of p and $AV^*(S)$ the furthest site anchored Voronoi diagram of S.

Lemma 4 ([8], Lemma1). *The furthest-site Voronoi diagram that corresponds to* $\mathcal{L}$ *is identical to the nearest-site Voronoi diagram that corresponds to* $\mathcal{L}^*$. *Moreover, if* $\mathcal{L}$ *is a semi-admissible system, then so is* $\mathcal{L}^*$.

In many cases the admissibility is not preserved when moving to the dual of the dominance system because the cells in a furthest site Voronoi diagram may be disconnected. In such case, the deterministic algorithm of [7] can not be used. However, in our case $\mathcal{L}^*$ fulfills the connectivity property.

Lemma 5. *Given a point* $p \in S$, *the cell* $AVR^*(p)$ *is a path-connected set.*

Proof. By definition, the cell $AVR^*(p)$ is the intersection of all regions $D(q,p)$ with $q \neq p$. In [2] have been proved that those regions are either convex or non-convex. Let us intersect both separately; call C the intersection of the convex regions and call D the intersection of the non-convex ones; see Figure 10 (a) and (b). Since C is the intersection of convex sets, C is path-connected. C is an unbounded region as depicted in Figure 10 (a).

Indeed, each bisecting curve is contained in a wedge determined by the half-lines of the bisecting curve and the origin (see section 2.1.). Therefore, the boundary of C is composed by two half-lines plus a sequence of pieces of bisecting curves. Region C is contained into a wedge C_1 given by two half-lines belonging to two bisecting curves.

Let us turn our attention to the non-convex region D. If wedges belonging to concave regions do not intersect C_1, then we know that intersection will be C, which is path-connected. Otherwise, some non-convex regions must intersect C. The only way for that intersection to give two or more connected components is that a bisecting curve in D intersects the pieces formed by bisecting curves in C at two or more points. We will show that situation cannot be possible.

Let us make a remark about C. Assume that C is the intersection of k bisecting curves. By Lemma 1, each pair of bisecting curves can only intersect each other once. Consider the half-lines of the wedges associated to each bisecting curve and number them in increasing order with respect to the angle as shown

in Figure 10 (a). One can see that, as we traverse the half-lines of the wedges, we find the sequence $\{1, 2, \cdots, k-1, k, 1, 2, \cdots, k-1, k\}$. Now, assume that a bisecting curve $B^{'}$ in D intersects the boundary of C at two different points, a and b. Then, the bisecting curve in C containing a intersects $B^{'}$ at two points. This is a consequence of the ordering in which bisecting curves in C intersect each other. This would be a contradiction with Lemma 1 and, therefore, the intersection gives only one connected component. Furthermore, the boundary of $C \cap D$ is composed by a convex chain plus a non-convex chain whose intersection consists of two points.

To end the proof, it is need to prove that $C \cap D = AVR^*(p)$ is path-connected set. Take two points a and b in $C \cap D$ and join them with the origin. Line segments $\overline{oa}$ and $\overline{ob}$ intersect the boundary of C at exactly one point each, say a_1, b_1, respectively. Line segments $\overline{aa_1}$ and $\overline{bb_1}$ are completely contained in $C \cap D$. On the other hand, we can go from a_1 to b_1 along the boundary of C (or at an infinitesimal distance from the boundary). This gives us a path fully contained connecting a and b and the claim is completely proved.

As a consequence of above results, $\mathcal{L}^*$ is a admissible system according to Klein's definition and the following result can be stated.

Theorem 2. *The furthest site anchored Voronoi $AVD^*(S)$ of a set of points S can be computed in $O(n \log n)$ time and $O(n)$ space.*

5 Applications

We next show how to use the $AVD(S)$ as a geometric structure for solving some facility location problems. The *obnoxious anchored bridge problem*, *OABP*, is stated as follows: *Let S be a set of n points in $\mathbb{R}^2 \setminus \{o\}$ and let $\mathcal{C}$ be a curve. Compute a line segment connecting o with a point x on $\mathcal{C}$ for which* $\min_{p \in S} d(p, \overline{ox})$ *is maximized.*

This problem arises in the transportation of obnoxious materials from a fixed depot to an existing route. See [4] for a recent survey on non-single facility location problems. Typically, in most applications, curve $\mathcal{C}$ will be an algebraic curve of constant degree, a trigonometric function or similar. Notice that, if $\mathcal{C}$ is a circle the problem becomes *the obnoxious anchored segment*, solved in [2].

Let $\overline{ox}$ be an optimal line segment for *OABP*. If $x \notin AVD(S)$ it is always possible to move the point x on the curve $\mathcal{C}$ without to decrease the minimum distance to the sites till an Voronoi edge is encountered. Thus, the following results can be stated.

Lemma 6. *There exists a point x^* which is the intersection between the curve $\mathcal{C}$ and the structure $AVD(S)$ such that the segment $\overline{ox^*}$ is a solution for the problem* OABP.

Theorem 3. *Once the $AVD(S)$ is given, the problem* OABP *can be solved in linear time and space.*

Notice that the furthest-site anchored Voronoi diagram can be also a suitable geometric structure for solving the center version of above problem. *The center anchored bridge problem*, *CABP*, asks for a line segment connecting o with a point x on $\mathcal{C}$ for which $\max_{p \in S} d(p, \overline{ox})$ is minimized. With similar arguments to the *OABP*, we can solve the *CABP* by restriction of bridges connecting the origin with the intersection points between the furthest-site anchored Voronoi and curve $\mathcal{C}$. Thus, we have the following result.

Theorem 4. *If the furthest-site anchored Voronoi is given, the problem* CABP *can be solved in linear time and space.*

6 Conclusion

We have introduced in this paper the anchored Voronoi diagram as an abstract Voronoi diagram. The bisecting curves are induced by the distance between a point and a line segment anchored at the origin. The concept of *circle* of an standard Voronoi diagram becomes the hippodrome in our context. The diagram AVD(S) has the empty circle properties: (1)two sites p and q share a Voronoi boundary if and only if there exists a hippodrome through p and q that does not contain any others sites in its interior and (2) a point x is a vertex of AVD(S) generated by p, q and r iff the hippodrome centered at $\overline{ox}$ and passing through p, q, r is empty. In this sense, the anchored Voronoi diagram can be considered as a suitable structure to solve both query and optimization problems when considering anchored line segments.

References

1. Aurenhammer F(1991) Voronoi diagrams: A survey of a fundamental geometric data structure. ACM Comput. Surv., 23, 345–405.
2. Barcia JA, Díaz-Báñez JM, Lozano A, Ventura I (2003) Computing an obnoxious anchored segment. Oper. Res. Letters, 31, 293–300.
3. Canny JF, Emiris I Z (1995) A General Approach to Removing Degeneracies. SIAM Journal of Computing, 24(3):650–664.
4. Díaz-Báñez JM, Mesa JA, Shöbel A (2004) Continuous location of dimensional structures. European J. of Operations Research, 152, 2004.
5. Edelsbrunner H, Seidel R (1986) Voronoi diagrams and arrangements. Discrete Computational Geometry, 1, 25–44.
6. Fortune S (1992) Voronoi diagrams and Delaunay trangulations. In Computing in Euclidean Geometry, D.-Z. Du and F.K. Hwang, eds, Lectures Notes Series on Comput. 1, World Scientific, Singapore, 193–233.
7. Klein R (1989) Concrete and Abstract Voronoi Diagrams. Lecture Notes in Computer Science, 400.
8. Mehlhorn K, Meiser S, Rasch R (2001) Furthest Site Abstract Voronoi diagrams. Int. J. of Comput. Geom. & Appl., 11, 6, 583–616.
9. Okabe A, Boots B, Sugihara K (1992) Spatial tessellations: concepts and applications of Voronoi diagrams, Wiley, Chichester, UK.

Implementation of the Voronoi-Delaunay Method for Analysis of Intermolecular Voids

A.V. Anikeenko[1], M.G. Alinchenko[1] ,V.P. Voloshin[1], N.N. Medvedev[1], M.L. Gavrilova[2], and P. Jedlovszky[3]

[1] Institute of Chemical Kinetics and Combustion SB RAS, Novosibirsk, Russia
nikmed@ns.kinetics.nsc.ru
[2] Department of Computer Science, University of Calgary, Calgary, AB, Canada
[3] Department of Colloid Chemistry, Eötvös Loránd University, Hungary

Abstract. Voronoi diagram and Delaunay tessellation have been used for a long time for structural analysis of computer simulation of simple liquids and glasses. However the method needs a generalization to be applicable to molecular and biological systems. Crucial points of implementation of the method for analysis of intermolecular voids in 3D are discussed in this paper. The main geometrical constructions - the Voronoi S-network and Delaunay S-simplexes, are discussed. The Voronoi network "lies" in the empty spaces between molecules and represents a "navigation map" for intermolecular voids. The Delaunay S-simplexes determine the simplest interatomic cavities and serve as building blocks for composing complex voids. An algorithm for the Voronoi S-network calculation is illustrated on example of lipid bilayer model.

1 Introduction

The Voronoi-Delaunay approach is well applicable to structural analysis of monatomic systems (computer models of simple liquids, amorphous solids, crystals, packings of balls). Geometrically these systems are represented as an ensemble of discrete points or spheres of equal radius, and the original mathematical premises of the method [8,28] are applicable for structural analysis of such systems [7,19,27]. Applying the method to molecular systems (molecular liquids, solutions, polymers, biological molecules) requires a modification of this classic data structure. The molecular systems usually consist of atoms of various radii; in addition, atoms in a molecule are connected via chemical bonds whose lengths are usually shorter than the sum of the atomic radii. Thus from a mathematical point of view a molecular system is an ensemble of balls of different radii some of which are partially overlapped.

One of the common problems in molecular systems analysis is determination of a region of space assigned to an atom [8,9]. The classical Voronoi polyhedron in 3D is suitable for this purpose in the systems of equal atoms but fails for a general case because its construction neglects atomic radii. It is well known that this problem can be solved using the additively weighted Voronoi diagram [20], where a measure of the distance between a point of space and a center of the i-th atom is defined as $\boldsymbol{y}=\boldsymbol{x}+W_i$, where $\boldsymbol{y}$ is an Euclidean distance between a pint in space and the center of the atom, and W_i is a *weight* of an i-th atom. This measure has a simple physical

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 217–226, 2004.
© Springer-Verlag Berlin Heidelberg 2004

interpretation. If value W_i is taken as a radius of i-th atom R_i then measure $\boldsymbol{y}$ represents a shortest Euclidean distance to its surface. Due to this fact this region is referred to as the Voronoi S-region in physics [4]. Note that Voronoi S-region can be defined not only for spheres, but also for physical bodies of other shape [14].

The next important physical problem is investigation of voids between molecules (cavities, pockets, channels). It differs from the calculations of region assigned to a given atom: any interatomic void is associated with a group of atoms, not with a single atom. In the case of monatomic systems the classical Delaunay simplexes are used for presentation of voids [6,25-27]. For molecular systems the Deluanay S-simplexes which present a dual constructions for the Voronoi S-region tessellation can be used analogously [4,17,18]. The Delaunay S-simplex is determined by centers of four atoms which are incidental to a common vertex of the Voronoi S-regions. This quadruplet of atoms gives the elementary (simplicial) cavity. Any complex void between atoms can be composed of such simplicial cavities. Connectivity of such simplexes can be studied using the Voronoi S-network, which is a network of edges and vertexes of the all Voronoi S-regions in the system. Interatomic voids can be analyzed using the Voronoi S-network and S-simplexes in frame of the ideology developed in this classical approach.

Despite the fact that the Voronoi-Delaunay constructions are well-studies, they are do not used to their full potential in physics or molecular biology, especially in three-dimensional environments. One of the reasons is the complexity of methodical and technical implementation of the method. In particular, after calculation of the Voronoi S-network and Deluanay S-simplexes one needs to use them to reveal voids and calculate their characteristics. In this paper, we address this problem and try to explain how the method can be implemented in order to be useful and make applications of Voronoi-Delaunay method more practical for physical applications. We also demonstrate that it can be an efficient method for studying large-scale 3D models.

2 Main Stages of Implementation of the 3D Model

2.1 Basic Geometric Concepts of the Voronoi-Delaunay Method

An initial construction of the approach is the *Voronoi S-region*: a region of space all points of which are closer to the *surface* of a given ball than to the *surfaces* of the other balls of the system, see Fig 1. For balls of the same size this region coincides obviously with the classical Voronoi polyhedron defined for the atomic centers, Fig.1a. However, the Voronoi S-region is not a polyhedron in general, for balls of different size its faces are pieces of hyperboloids (Fig.1b). The most important peculiarity of the S-region is that it determines naturally a region of space assigned to a given ball (molecule). Jointly, the edges and vertices of Voronoi S-regions form the Voronoi S-network of a given system (thick lines in Fig.1c).

It is known that Voronoi S-regions constructed for all atoms of the system form a partitioning which covers the space without overlapping and gaps [5,9,11,12,18,20]. This *Voronoi S-tesselation* divides the atoms of the system, similarly to the classical Voronoi tessellation, into the quadruplets of atoms (*the Delaunay S-simplexes*), representing elementary cavities between the atoms (see Fig. 2).

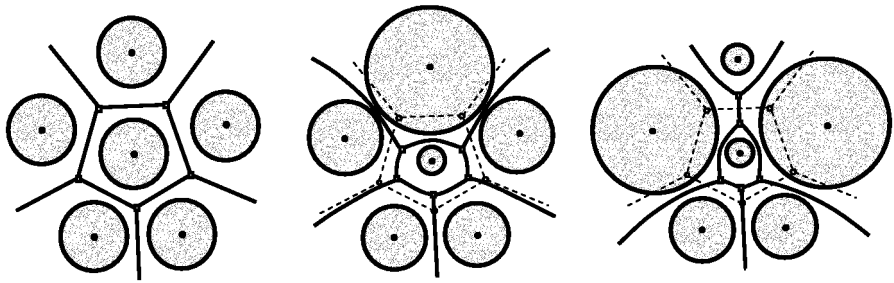


Fig. 1. 2D-illustration of the Voronoi S-regions in systems of atoms of equal and different radii.

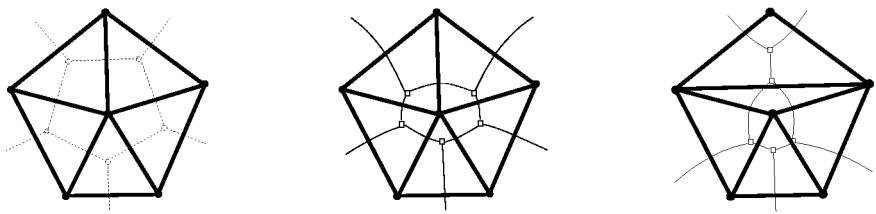


Fig. 2. The Delaunay S-simplexes (thick lines) for configurations in Fig.1. They can coincide with the classical Delaunay simplexes as in (a) and (b) or be different (c) which depends on radii of atoms. Thin lines show the Voronoi S-network.

The set of all vertices and edges of the Voronoi S-regions determines *the Voronoi S-network*. Each vertex (site) of the Voronoi S-network is the center of *the interstitial sphere*, which corresponds to one of the Delaunay S-simplexes. Each edge (bond) is a fairway passing through the *bottleneck* between three atoms from one site to the neighboring one.

2.2 Basic Data Structure for Representation of the Voronoi S-Network

To work with any network we should know coordinates of the network sites and their connectivity. In addition, to calculate characteristics of voids one needs radii of interstitial spheres and radii of bottle-necks. Let *array_D* contain the coordinates of network sites. An order in which the sites are recorded in this array defines the numbering of the sites. Let *array_*R_i contain the radii of interstitial spheres. Each sphere corresponds to one of the sites of the network. *Array DD* establishes the connectivity of network sites. By that it determines the bonds of the network. Each bond defines the bottleneck between a pair of sites. It is useful to have a special *array_Rb* which contains the radii for all bottlenecks; they are needed for analysis of complex voids. Finally, explicit information is desirable for work with S-simplexes. To this end, the simplest way is to create *array_DA* representing a table of incidence of the network sites and the numbers of atoms relating to the corresponding sites. All this information is calculated at calculation of the Voronoi S-network. The empty

volume of all S-simplexes should be calculated and recorded in an *array_Ve*. This information is enough to start the analysis of voids.

Note the problem of overlapping atoms can be solved easily for analysis of voids. Since each bond of the Voronoi S-network is a locus of the points equidistant from the surfaces of the nearest three balls, this locus does not change if we decrease (or increase) the radii of the balls by the same value *d*. Similarly, the site (the common vertex of the Voronoi S-regions) does not change its position if the radii of the corresponding four balls are changed by the same value. Thus we construct a reduced system by decreasing the radii of all balls of the initial system by some constant value *d* to avoid the overlapping of atoms. Then we construct the Voronoi S-network using the algorithm for constructing the S-network for the system of non-overlapping balls.

The required arrays that determine the Voronoi S-network (the coordinates of network sites, the table of connectivity, the table of the incidence of sites and atoms) fully coincide for the initial and reduced systems. The values of the radii of interstitial spheres and bottlenecks for the initial system are obviously different from the corresponding values of the reduced system on the constant value *d*.

2.3 Determination of Interatomic Voids on the Voronoi S-Network

Empty space in a 3D-system of atoms is a complex singly connected system confined by spherical surfaces of atoms. Any interatomic void to be distinguished is a part of this system and it depends on the detection criterion. A physical way of defining voids is through the value of the radius of a probe (test sphere), which can be located in a given void. The number, size and morphology of the voids depend on the probe radius: some most spacious cavities represent voids for large probe, and almost the entire interatomic space is accessible for small one.

A simple but important characteristic of interatomic space is a set of interstitial spheres. These spheres represent real empty volume between atoms. The values of their radii indicate the scale of voids in a system. A more comprehensive analysis of voids and interatomic channels requires knowing the system of bottlenecks, i.e., the analysis of the bonds of the Voronoi S-network. If a probe can be moved along the bond then the network sites at the ends of the bond are sure to be also accessible for this probe [18]. Thus, the regions accessible for a given probe can be found by distinguishing the bonds whose bottleneck radius exceeds a given value, see Fig.3. The clusters consisting of these bonds represent the fairways (skeletons) of the regions along which a given probe can be moved. Distinguishing bonds on the Voronoi S-network using bottlenecks is called *Rb-coloring* of the network [26,16,18].

Representation of voids by clusters of colored bonds is highly descriptive for illustrating the locations of complex voids inside the model. However, to perform a deeper physical analysis of the voids, their volumes should be calculated. This can be performed with the help of the Delaunay S-simplexes. Knowing the sites of the Voronoi S-network involved in a given cluster, we know all S-simplexes composing this void. The union of the empty volumes of these S-simplexes provides the “body” of the void to be found, Fig. 3 (right). The rest of the empty space in the model is inaccessible for such probe. The proposed representation of voids provides a quantitative basis for analyzing the various characteristics of voids

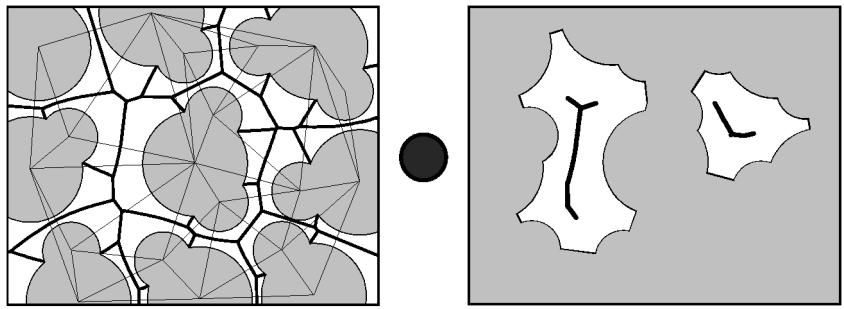


Fig. 3. Left: The Voronoi S-network (thick lines) and Delaunay S-simplexes (thin lines) for a molecular system. Each bond of the network is a fairway passing through the bottleneck between atoms. Right: Voids in the system accessible for a probe shown as a disk between figures. The fairways of the voids (thick lines) are clusters of the Voronoi S-network bonds which bottleneck radii is greater then radius of the probe.

2.4 Representation of Voids by Spherocylinders and Calculation of Empty Volume

The voids accessible for relatively large probes (i.e. of the order of atomic size) are more interesting for physicists since these voids might play an important role in the mechanism of the diffusion of small molecules. The radii of the probes used fall in the range between 1.0Å and 1.6Å. Preliminary analysis shows that for molecular systems voids corresponding to such probes are usually rather compact. (A complex, branching structure of the voids starts to appear for considerably smaller probes). Although the shape of the compact voids is not simple, their main characteristics can be described by just a few parameters, such as their length, width and orientation, implying that these voids can be represented by bodies of rather simple shape in order to make their detailed analysis mathematically feasible.

For our analysis, we suggest to represent voids by spherocylinders [1] (i.e., cylinders covered by hemispheres of the same radius at the two basic circular faces). We calculate these parameters directly, by means of the "inertia tensor" of the void instead of artificially "fitting" them. The inertia tensor of a void is calculated using the cluster of bonds and sites on the Voronoi S-network representing a given void. The fictitious "mass", equal to the value of the empty volume of the corresponding Delaunay S-simplex, is assigned to each site of the cluster. Thus, the volume of the void is concentrated on the S-network sites, and hence the continuous body of complex shape of the void is represented by a system of a finite number of "massive" points, for which the inertia tensor can be readily calculated. The axis along which the principal value of the inertia tensor is minimal indicates the direction of the largest extension of the void. It is taken as the axis of the required spherocylinder. To calculate the length of the spherocylinder L (i.e., the length of its axis in the cylindrical part), all the sites of the cluster are projected to this axis, and the mean square deviation of these projections from the centre of the fictitious mass of the cluster (lying always on this axis) are calculated. Finally, the radius R of the spherocylinder is unambiguously determined from the condition of the equality of the volumes of the spherocylinder and the void. This condition can be written simply as

$$V_{void} = R^2\pi\left(L + \frac{4}{3}R\right), \quad (1)$$

where V_{void} is the volume of the void, determined as the sum of the empty volumes of the composing Delaunay S-simplexes.

At the first glance, the empty volume inside simplexes seems to be easily calculated analytically as the volume of the whole simplex minus the volume of the parts of its own atoms composing this simplex. However, the simplex often involves "alien" atoms assigned to other simplexes [23,18]. It is more difficult to take into account the volume occupied by these atoms. Moreover, there can be several alien atoms incoming in a given simplex. Additional problems arise for the case of overlapping atoms, which should be taken into account to correctly compute the empty volume. Thus, it is rather difficult to derive an analytical formula for calculation of the empty simplex volume. This, however, can be done numerically. To this end, we fill the simplex with sampling points (randomly or regularly) and determine the fraction of points outside the atoms. Implementation of this idea can be rather efficient because the list of atoms that can enter in the simplex is readily defined by the Voronoi S-tessellation. Note that under certain conditions some Delaunay S-simplexes can cover each other. Such covering of S-simplexes can result in error during calculation of the volume of complex voids. Fortunately, this possibility can be ignored for molecular systems. To verify this, we had compared the sum of the volumes of all Delaunay S-simplexes with total volume of the model. The difference was found is negligibly small for our models (hundredth of a percent).

3 The Improved Algorithm for 3D Voronoi S-Network Calculation

There were several attempts to implement an algorithm for additively weighted Voronoi construction. However a detailed investigation of the problem has been made only for the case of 2D [9,11,12,20]. The 3D applications are restricted to the Voronoi S-regions (additively weighted Voronoi cell) [10,22]. In our earlier papers [4] we used our previous version of the algorithm for the Voronoi S-network calculation, but it was not very efficient for large models. A specific algorithm for numerical calculations of S-network for straight lines and spherocylinders was realized in [14], which can be also applied to spherical particles but it is much slower. Here we present our method which is efficient for large models and specialized for investigation of voids in complex molecular systems.

The main idea is simple and based on technique proposed many years ago for calculation of the 3D Voronoi polyhedra [24,15], where starting from a Voronoi vertex (site), the neighboring sites are calculated consecutively for every face of the polyhedron. A difference now is only that the other formulas for calculation of the coordinates of sites are used, see e.g. [9,18,20]. To calculate a new site we involve in calculation only limited number of atoms in the neighborhood of a given site. If we know these neighbours, the CPU time for calculation of a site does not depend on a total number of atoms in the model. Using linked-list based structure, which establishes a correspondence among coordinates and numbers of atoms, we immediately reestablish atoms which are close to a given point (e.g. to a site of the network) [2]. This improvement makes construction of Voronoi network much faster.

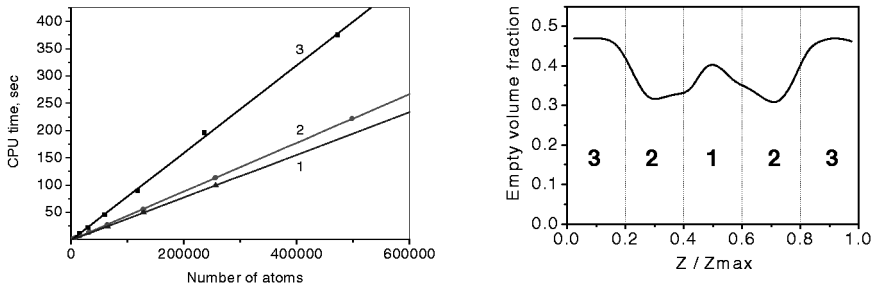


Fig. 4. Left: CPU time for Voronoi S-network computation as a function of the number of atoms in the model. Right: The profile of the fraction of empty volume across the simulated lipid bilayer along the membrane normal axis *Z*. Dashed vertical lines show the division of the system into three separate regions according to the behavior of this profile

For illustration of efficiency of the algorithm we carried out some tests (Fig. 4 left). A PC with Intel processor P4 with 1700 MHz and RAM 256 was used. Three types of models were tested: dense non-crystalline packing of equal balls (curve 1), dense disordered packing of balls with radii 1 and 0.5 in fraction of one-to-one (curve 2), and a molecular system on the base of the model of lipid bilayer in water (curve 3). Starting configurations of the models for all types contained about 10000 atoms in boxes with periodic boundary conditions. Enlargement of the models was made by replication of starting configuration into bigger box according the periodic boundary condition. Calculation of the Voronoi S-network implies creation of the arrays D, DD, DA, Ri, Rb and their recording in hard disk of the computer. Different types of models demonstrate different CPU time (what depends of the structure of models), but all of them demonstrate clear lineal dependence on number of atoms.

4 Application of Model to Analysis of Lipid Bilayers

We illustrate application of the method to a computer model of the fully hydrated DMPC bilayer as obtained from a recent all-atom Monte Carlo simulation, Fig. 5. Each of the two membrane layers contain 25 DMPC molecules, described by the CHARMM22 force field optimized for proteins and phospholipid molecules and the bilayer is hydrated by 2033 water molecules. The sample analyzed consists of 1000 independent configurations, each of them saved after performing 10^5 new Monte Carlo.

In analyzing the distribution and properties of the voids in the model we have first determined the fraction of the empty space across the membrane. The resulting profile along the membrane normal axis *Z* is shown in Fig. 4 (right). As seen, three different membrane regions can be clearly distinguished according to the behaviour of this profile. Region 1, in the middle of the membrane, is characterized by a relatively large fraction of the empty volume, which is considerably lower in the adjacent region 2. Finally, in region 3, located apart from the lipid bilayer, the fraction of the empty space is the highest in the entire system. These regions, marked also in Fig. 4 (right), roughly coincide with the region of the hydrophobic lipid tails, the dense region of the hydrated zwitterionic headgroups and the region of bulk-like water, respectively.

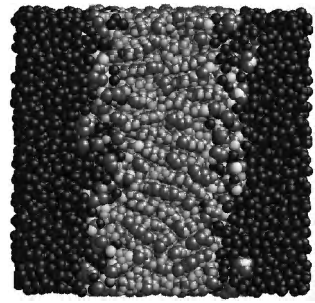


Fig. 5. 3D configuration of the DMPC lipid bilayer in a box with periodic boundary conditions.

5 Experimental Analysis

In the analysis, we have determined the voids using different values of the probe radius between 1.0Å and 1.6Å, with an increment of 0.1Å. In this way, the criterion of void detection has been varied in the analyses, allowing us a more reliable characterization of the properties of the voids. For quantitative characterization of the voids we depicture them as spherocylinders, see section 2.5. In the following, characteristics of the spherocylinders (i.e., length, radius, volume and orientation) are studied as a function of the probe radius in three separate regions of the membrane.

The dependence of the mean values of the length L of the spherocylinders on the probe radius in the three membrane regions are shown in Fig. 6 (left). As is seen, the observed length of the voids is clearly different in the three different parts of the membrane, and this difference is preserved for all probe radii used, showing that this finding is independent from the void detection criterion. It should be noted that the longest spherocylinders, and hence the most elongated voids are found in region 1, i.e., at the middle of the membrane, whereas the largest fraction of the empty volume occurs in the region of bulk-like water (region 3, see Fig.6 (left)). This finding indicates that the empty volume is distributed considerably more uniformly in the aqueous region than in the hydrocarbon phase of the bilayer.

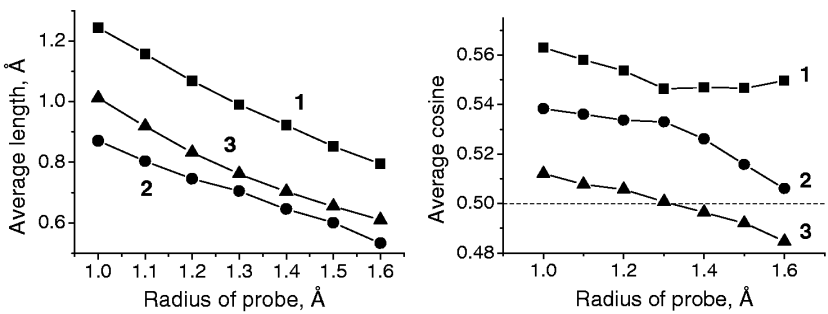


Fig. 6. An average length L of the spherocylinders representing the voids (left) and an average cosine of the angle α formed by the bilayer normal axis Z with the main axis of the spherocylinders representing the voids (right), as a function of the probe radius. Squares: region 1 (hydrocarbon tails), circles: region 2 (headgroups), triangles: region 3 (bulk-like water).

In analyzing the orientation of the voids we have calculated the mean cosine of the angle α formed by the main axis of the spherocylinder and the bilayer normal axis *Z*. Isotropic orientation of spherocylinders results in the mean cosine value of 0.5, whereas for preferential orientations perpendicular and parallel to the plane of bilayer the inequalities $cos\alpha>0.5$ and $cos\alpha<0.5$, respectively, hold. The dependence of the mean value of $cos\alpha$ on the probe radius in the three separate membrane regions is shown in Fig. 6(right). Observe that the mean cosine value is clearly larger than 0.5 in regions 1 and 2, being larger in region 1 than 2 for all probe sizes used. Thus, the preferred orientation of the pores is clearly perpendicular to the plane of the membrane in the region of the hydrophobic tails, and also, in a smaller extent, in the region of the headgroups. This finding reflects the fact that the arrangement of the voids located between the lipid tails follows the preferred arrangement of these tails.

6 Conclusion

Application of the Voronoi–Delaunay method to molecular system modeling and analysis are discussed in the paper. The methodological aspects of Voronoi network consturtion in 3D, determination of intermolecular voids, and calculation of some physical characteristics of voids are presented. The method illustrated on the experimental analysis of the Monte Carlo models of fully hydrated DMPC bilayer. It provides a detailed overview of pitfalls and a handy solution to analysis of general behaviour of the interatomic voids in such systems.

Acknowledgements. This work was supported by the RFFI (grant 01-03-32903 and 04-03-32283), INTAS (grant 01-0067), OTKA (grant F038187) and CRDF (grant NO-008-X1), University of Calgary Starter Grant, and NSERC Grant.

References

1. Alinchenko M.G, A.V.Anikeenko, N.N.Medvedev, V.P.Voloshin, P. Jedlovszky. Morphology of voids in molecular system. *J.Chem.Phys.* in prep. (2004).
2. Allen M.P., Tidesley D.J. Computer simulation of liquids Claredon Press, 1987.
3. Angelov B., Sadoc J.F., Jullien R, SoyerA, Mornon J.P., Chomilier J. Nonatomic solvent-driven Voronoi tessellation of proteins: An open tool to analyze protein folds. *Proteins-Structure Function and Genetics,* vol.49 , no.4, p. 446–456, 2002.
4. Anishchik S. V and Medvedev N. N., Three dimensional Apollonian packing as a model for dense granular systems. *Phys. Rev. Lett*., vol.75, no.23, p. 4314–4317
5. Aurenhammer F., Voronoi Diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv*., Vol.23, 1991, pp. 345–405.
6. Bryant S., Blunt M, Prediction of relative permeability in simple pore media. *Phys.Rev.A,* Vol. 46, No.6, 1992, pp. 2004–2011.
7. Delaunay simplexes in the structure analysis of molecular-dynamics-simulated materials. Phys.Rev.B, Vol. 57, No**.** 21, 1998, pp. 13448–13458.
8. Delaunay B.N., Sur la sphere vide. *Math. Congress Aug.11-16,1924*, pp. 695–700.
9. Gavrilova M., *Proximity and Applications in General Metrics,* Ph. D. Thesis, The University of Calgary, Dept. of Computer Science, Calgary, AB, Canada, 1998.

10. Goede A., Preissner R., Froemmel. Voronoi Cell: New method for allocation of space among atoms: elimination of avoidable errors in calculation of atomic volume and density., *J.Comp.Chem*. Vol. 18, 1997, pp. 1113.
11. Kim D.-S., Kim D, Sugihara K. Voronoi diagram of a circle set from Voronoi diagram of a point set: I. Topology. *CAGD* 18(6): 541 (2001).
12. Karavelas M., and Yvinec M. Dynamic Additively Weighted Voronoi Diagrams in 2D. *In Proc. 10th European Symposium on Algorithms*, 2002, pp. 586–598,
13. Liao Y.C., Lee D.J., He P.J. Microstructural description of packed bed using Voronoi polyhedra. *Powder technology*, Vol. 123, No. 1, 2002, pp. 1–8.
14. Luchnikov V.A., Medvedev N.N., Oger L, Troadec J.-P. The Voronoi-Delaunay analysis of voids in system of non-spherical particles *Phys.Rev. E* 59(6):7205, 1999
15. Medvedev N.N. Algorithm for three-dimensional Voronoi polyhedra. *J. Comput. Physics.* Vol..67, 1986, pp. 223–229.
16. Medvedev N.N., Naberukhin Yu.I. *J.Phys.A: Math.Gen.*, 21, L247 (1988).
17. Medvedev N.N., *Computational porosimetry,* pp.164-175, in the book Voronoi's impact onmodern science, ed. P.Engel, H.Syta, Inst.Math., Kiev, 1998.
18. Medvedev N.N.. *Voronoi-Delaunay method for non-crystalline structures*. SB Russian Academy of Science, Novosibirsk, 2000, 209, (in Russian).
19. Oger L, Gervois A, Troadec J-P., Rivier N: Voronoi tessellation of packing of spheres: topological correlation and statistic *Philosoph.l Mag* 74:177–197, 1996
20. Okabe A, Boots B, Sugihara K, and Chiu S.N., *Spatial Tessellations: Concepts and applications of Voronoi diagrams (Chichester, John Wiley),* 2000.
21. Richards F.M., Calculation of molecular volumes and areas for structures of known geometry. *Methods in Enzymology*, Vol.115, 1985, pp. 440–464.
22. Richard P, .Oger L, Troadec J.P., and Gervois A. A model of binary assemblies of spheres *Eur.Phys. E.,* Vol. 6, 2001, pp. 295–303.
23. Sastry S, Corti D.S., Debenedetti P.G., Stillinger F.H. Statistical geometry of particle packings. *Phys.Rev.E.* Vol.56, No. 5, 1997, pp. 55245532.
24. Tanemura M., Ogawa T., Ogita N. A new algorithm for three-dimensional Voronoi tessellation. J. Comput. Phys. Vol.51, 1983, pp. 191–207.
25. Thompson K.E., Fogler H.S., Modelling flow in disordered packed bed from pore-scale fluid mechanics. *AIChE Journal*, Vol.43, 1997, pp. 1377.
26. Voloshin V.P., Naberukhin Yu.I., Empty interatomic space in computer models of simple liquis and amorphous solids, J.Phys: Condens. Matter, 5, pp. 5685, 1993
27. Voloshin V.P., Beaufils S., and Medvedev N.N. Void space analysis of the structure of liquids. J. of Mol. Liq. Vol. 96-97, 2002, pp. 101–112.
28. Voronoi G.F. Nouvelles applications des paremetres continus a la theorie des formes quadratiques. *J.Reine Andew.Math*., Vol.134, 1908, pp. 198–287.

Approximation of the Boat-Sail Voronoi Diagram and Its Application

Tetsushi Nishida and Kokichi Sugihara

Department of Mathematical Informatics,
University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{nishida,sugihara}@mist.i.u-tokyo.ac.jp

Abstract. A new generalized Voronoi diagram, called a boat-sail Voronoi diagram, is defined on the basis of the time necessary for a boat to reach on water surface with flow. The problem of computing this Voronoi diagram is reduced to a boundary value problem of a partial differential equation, and a numerical method for solving this problem is constructed. The method is a modification of a so-called fast marching method originally proposed for the eikonal equation. Computational experiments show the efficiency and the stableness of the proposal method. We also apply our equation to the simulation of the forest fire.

1 Introduction

The Voronoi diagram is one of the most important data structures in computational geometry, and has been studied extensively from both algorithmic and application points of view [4,9]. The Voronoi diagram has also been generalized in various directions. One of typical directions is the generalization of the distance. The most fundamental Voronoi diagram is defined according to the Euclidean distance, while many generalized Voronoi diagrams are generated by replacing the Euclidean distance with other distances. [1,2,5,6,12]

As one of the generalizations, we introduce the boat-sail Voronoi diagram [7]. Suppose that we want to travel on the surface of water with a boat. If there is no flow of water, the boat can move in any direction at the same maximum speed. If the water flows, on the other hand, the speed of the boat is anisotropic; the boat can move faster in the same direction as the flow, while it move only slowly in the direction opposite to the flow direction. Modeling this situation, we can introduce the boat-sail distance and can obtain the boat-sail Voronoi diagram.

In order to compute this boat-sail distance and the associated Voronoi diagram, we reduced the problem to a boundary value problem of a partial differential equation [7]. This idea is the same as the idea for reducing the problem of computing the Euclidean distance to a boundary value problem of the eikonal equation [3,10]. Hence, our formulation can be considered a generalization of the eikonal equation.

In this paper, we firstly propose a new scheme to stably solve our partial differential equation. This scheme is the extension of the first marching method

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 227–236, 2004.
© Springer-Verlag Berlin Heidelberg 2004

[10] which was originally proposed for the eikonal equation. Secondly, we consider to apply our equation to the simulation of the forest fire.

In Section 2, we introduce a mathematical model for the boat-sail distance and the associated Voronoi diagram. In Section 3, we derive a partial differential equation for representing the boat-sail distance. In Section 4, we construct a new scheme for computation of the boat-sail distance, which is a combination of the fast marching method and the finite-element method. In Section 5, we present numerical examples, and finally, we give concluding remarks in Section 6.

2 Boat-Sail Distance and the Associated Voronoi Diagram

Let $\Omega \subset \mathbf{R^2}$ denote a two-dimensional domain with an (x, y) Cartesian coordinate system, and let $f(x, y) \in \mathbf{R^2}$ be a two-dimensional vector given at each point (x, y) in Ω. A physical interpretation is that Ω corresponds to the surface of water and $f(x, y)$ represents the velocity of the water flow. Hence, we call $f(x, y)$ the flow field. We assume that $f(x, y)$ is continuous in Ω.

Consider a boat that has the maximum speed F in any direction on the still water. Let Δt denote a short time interval. Suppose that the driver tries to move the boat at speed F in the direction v_F, where v_F is the unit vector, and hence the boat will move from the current point p to $p + \Delta t F v_F$ in time Δt if there is no water flow, as shown by the broken arrow in Fig 1. However, the flow of water also displaces the boat by $\Delta t f(x, y)$, and hence the actual movement Δu of the boat in time interval Δt is represented by $\Delta u = \Delta t F v_F + \Delta t f(x, y)$.

Consequently, the effective speed of the boat in the water flow is given by

$$\left| \frac{\Delta u}{\Delta t} \right| = |F v_F + f(x, y)| \,. \tag{1}$$

We assume that F is large enough to satisfy the condition $F > \max_{(x,y)\in\Omega} |f(x, y)|$.

Let p and q be two points in Ω, and let $c(s) \in \Omega$ denote a curve from p to q with the arc-length parameter s $(0 \le s \le \overline{s})$ such that $c(0) = p$ and $c(\overline{s}) = q$. Then, the time, say $\delta(c, p, q)$, necessary for the boat to move from p to q along the curve $c(s)$ with the maximum speed is obtained by

$$\delta(c, p, q) \equiv \int_0^{\overline{s}} \left| \frac{\Delta t}{\Delta u} \right| \mathrm{d}s = \int_0^{\overline{s}} \frac{1}{|F v_F + f(x, y)|} \mathrm{d}s. \tag{2}$$

Let C be the set of all paths from p to q. We define $d(p, q)$ by

$$d(p, q) \equiv \min_{c\in C} \delta(c, p, q). \tag{3}$$

That is, $d(p, q)$ represents the shortest time necessary for the boat to move from p to q. We call $d(p, q)$ the *boat-sail distance* from p to q.

Next, we define a generalized Voronoi diagram with respect to the boat-sail distance. Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of n points, called boat harbors, in Ω. For $p_i \in P$, we define region $R(P; p_i)$ by

$$R(P; p_i) \equiv \bigcap\nolimits_{j \neq i} \{p \in \Omega \mid d(p_i, p) < d(p_j, p)\}. \tag{4}$$

$R(P; p_i)$ represents the set of points which the boat at harbor p_i can reach faster than any other boats. The domain Ω is partitioned into $R(P; p_1)$, $R(P; p_2)$, $\cdots$, $R(P; p_n)$ and their boundaries. This partition is called the *Voronoi diagram for the boat-sail distance* or the *boat-sail Voronoi diagram* for short.

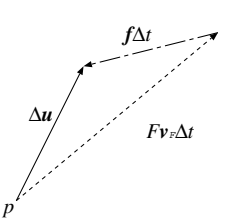


Fig. 1. Relation among the actual movement Δu, the water flow f and the boat velocity Fv_F

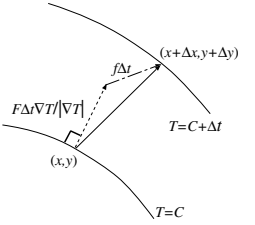


Fig. 2. Decomposition of the movement of a boat

3 Reduction to a Boundary Value Problem

Suppose that we are given the flow field $f(x, y)$ and the point $p_0 = (x_0, y_0)$ of the boat harbor in Ω. Let $T(x, y)$ be the shortest arrival time at which the boat departing p_0 at time 0 can reach the point $p = (x, y)$, that is, $T(x, y) \equiv d(p_0, p)$. In this section, we derive the partial differential equation that should be satisfied by the unknown function $T(x, y)$.

Let C be an arbitrary positive constant. The equation $T(x, y) = C$ represents a curve, any point on which can be reached in time C by the boat departing p_0 at time 0. As shown in Fig. 2, assume that the boat moving along the shortest path passes through the point (x, y) at time C and reaches the point $(x + \Delta x, y + \Delta y)$ at time $C + \Delta t$, where Δt is positive and small. Hence, in particular, we get

$$T(x + \Delta x, y + \Delta y) - T(x, y) = \Delta t. \tag{5}$$

If there is no flow, the shortest path should be perpendicular to the curve $T = C$, and hence, the progress of the boat during time interval Δt is represented by $F\frac{\nabla T}{|\nabla T|}\Delta t$. On the other hand, the displacement of the boat caused by the flow is $f\Delta t$. Hence, the total motion of the boat is represented by

$$F\frac{\nabla T}{|\nabla T|}\Delta t + f\Delta t. \tag{6}$$

Let us denote $T_x \equiv \frac{\partial T}{\partial x}$ and $T_y \equiv \frac{\partial T}{\partial y}$. Also let $g(x, y)$ and $h(x, y)$ denote the first and second components of $f(x, y)$. Then from the equation (6), we get

$$\Delta x = F\frac{T_x}{|\nabla T|}\Delta t + g\Delta t, \quad \Delta y = F\frac{T_y}{|\nabla T|}\Delta t + h\Delta t. \tag{7}$$

Hence, we get

$$\begin{aligned} T(x + \Delta x, y + \Delta y) &= T(x, y) + T_x\Delta x + T_y\Delta y + \mathrm{O}((\Delta x)^2 + (\Delta y)^2) \\ &= T(x, y) + T_x(F\frac{T_x}{|\nabla T|} + g)\Delta t + T_y(F\frac{T_y}{|\nabla T|} + h)\Delta t + \mathrm{O}(\Delta t^2). \end{aligned}$$

Substituting this equation in equation (5), we get

$$F|\nabla T| = 1 - \nabla T \cdot f. \tag{8}$$

This is the partial differential equation that should be satisfied by the arrival time $T(x, y)$.

In the next section, we consider how to solve this partial differential equation numerically, together with the boundary condition

$$T(x_0, y_0) = 0. \tag{9}$$

4 FEM-Like Fast Marching Method

Our equation has the property that the arrival time $T(x, y)$ is monotone increasing as we move along the shortest paths starting at p_0. A typical equation of this type is the eikonal equation [10], which can be solved efficiently and stably by the fast marching method [10].

We, however, recognize from numerical experiments that the fast marching method did not work for our equation [7,8]. Hence, we propose a new scheme by modifying the fast marching method. In this section, we briefly overview it; for further details, we refer to [8]

4.1 FEM-Like Differences

In Ω, we place grid points $(x_i, y_j) = (i\Delta x, j\Delta y)$, $i, j = 0, \pm 1, \pm 2, \cdots$, where Δx and Δy are small constants and i and j are integers. For each grid point (x_i, y_j), we associate $T_{ij} = T(x_i, y_j)$. $T_{00} = T(x_0, y_0) = 0$ because of the boundary condition (9), while all the other T_{ij}'s are unknown variables.

Starting with the neighbors of (x_0, y_0), we want to compute T_{ij}'s grid by grid from smaller values to larger values. Hence, we use the modified upwind differences which we explain as follows.

In this section, we propose the extension of the second order upwind difference [10]. Considering grid points on a triangular element shown in Fig. 3, we can derive the differences at a target point from these grid points, where the target

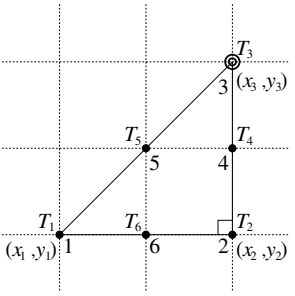


Fig. 3. An examples of triangular finite element

point is represented by the double circles. Note that, for a target point, there are eight triangles. Fig. 3 shows an example; the other seven triangles can be obtained by rotating this triangle by $\pi/2$, π and $3\pi/2$ around the target point, and by mirroring them with respect to the horizontal and the vertical lines passing through the target point.

Let the coordinates of nodes 1, 2 and 3 be (x_1, y_1), (x_2, y_2) and (x_3, y_3), respectively and nodes 4, 5 and 6 the middle points of the edges. Also, let $T_1, T_2, \ldots, T_6$ be the values at the nodes $1, 2, \ldots, 6$, respectively. Then, the interpolation function T which represents the value at point (x, y) in the triangular element is represented by

$$T(x,y) = 4T_4\phi_2(x,y)\phi_3(x,y)+4T_5\phi_3(x,y)\phi_1(x,y)+4T_6\phi_1(x,y)\phi_2(x,y)$$
$$+T_1\phi_1(x,y)(2\phi_1(x,y)-1)+T_2\phi_2(x,y)(2\phi_2(x,y)-1)+T_3\phi_3(x,y)(2\phi_3(x,y)-1)$$

where $\phi_1(x, y)$, $\phi_2(x, y)$ and $\phi_3(x, y)$ are the area coordinate functions:

$$\phi_i(x,y) = \frac{1}{D}(a_i + b_i x + c_i y),$$

where

$$D = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \text{ and } \begin{cases} a_1 = x_2y_3 - x_3y_2,\ b_1 = y_2 - y_3,\ c_1 = x_3 - x_2, \\ a_2 = x_3y_1 - x_1y_3,\ b_2 = y_3 - y_1,\ c_2 = x_1 - x_3, \\ a_3 = x_1y_2 - x_2y_1,\ b_3 = y_1 - y_2,\ c_3 = x_2 - x_1. \end{cases}$$

Partially differentiating this interpolation function and substituting (x_3, y_3) into the partial derivatives obtained now, we can get the the values of the partial derivatives $\frac{\partial T}{\partial x}(x_3, y_3)$ and $\frac{\partial T}{\partial y}(x_3, y_3)$ at the node 3. Next, let us respectively replace (x_3, y_3) and T_3 by (x_i, y_j) and T_{ij}. Then we get the value of the partial derivatives at each grid point (x_i, y_j):

$$D^x_{ij}T \equiv \frac{3b_3T_{ij} + 4(b_2T_4 + b_1T_5) - b_1T_1 - b_2T_2}{D}, \tag{10}$$

$$D^y_{ij}T \equiv \frac{3c_3T_{ij} + 4(c_2T_4 + c_1T_5) - c_1T_1 - c_2T_2}{D}. \tag{11}$$

We call (10) and (11) the *second order FEM-like differences*.

However, if $T_1 < T_5$ or $T_2 < T_4$, the second order differences cannot be used [11]. Then we have to use the first order differences. The first order differences can be derived in a similar manner as the second order differences [8] . In what follows, we derive our scheme on condition that the second order differences are available.

Let us define g_{ij} and h_{ij} by $g_{ij} = g(x_i, y_j)$ and $h_{ij} = h(x_i, y_j)$, respectively. We replace ∇T by $(D^x_{ij}T, D^y_{ij}T)$ and f by (g_{ij}, h_{ij}) in our equation (8). Then we obtain the difference version of the equation:

$$F^2\{(D^x_{ij}T)^2 + (D^y_{ij}T)^2\} = (1 - (D^x_{ij}T)g_{ij} + (D^y_{ij}T)h_{ij})^2. \tag{12}$$

Solving this equation, we obtain the unknown arrival time T_{ij} at (x_i, y_j) from smaller arrival times around it.

4.2 Choice of a Triangle

For each target point, there are eight triangles: the triangle in Fig. 3 and its rotated/mirrored version. The next question is which triangle we should choose for the most precise computation. The best triangle is what includes the shortest path to p_3. Consider the triangle $p_1p_2p_3$ shown in Fig. 4. Let n_1 and n_2 be outer normal vectors for the edges p_1p_3 and p_2p_3. Also, let n'_1 (n'_2) be the vector directed from $p_1(p_2)$ to p_3. Then, the triangle includes the shortest path to p_3 if and only if the direction of the shortest path is between n'_1 and n'_2. Hence, from equation (6), this condition can be expressed by

$$\left(F\frac{\nabla T}{|\nabla T|} + f\right) \cdot n_1 \geq 0 \text{ and } \left(F\frac{\nabla T}{|\nabla T|} + f\right) \cdot n_2 \geq 0. \tag{13}$$

Therefore, we find the triangle that satisfies the condition (13), and use it for generating the equation (12).

Thus, we can solve our partial differential equation (8) by solving the finite difference equation (12) associated with the triangle chosen by this strategy. We call this method the *FEM-like scheme*.

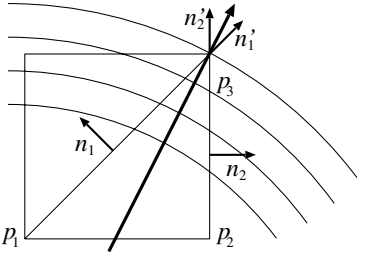


Fig. 4. Relation between the shortest path and the best triangle

4.3 Algorithm

Our overall method is similar to the Dijkstra method. We consider the grid structure the graph in which the vertices are grid points and the edges connect each grid point with its eight neighbors. We start with the boat harbor at which $T_{ij} = 0$, and compute T_{ij}'s one by one from the nearest grid point. The only difference from the Dijkstra method is that the quadratic equation (12) is solved to obtain T_{ij}.

In the next algorithm, the grid points are classified into three groups: "known" points, "frontier" points and "far" points. The "known" points are points at which the values T_{ij} are known. The "frontier" points are points that are not yet known but are the neighbors of the "known" points. The "far" points are all the other points.

Suppose that there are n boat harbors, and they are numbered $1, 2, \cdots, n$. Let S_{ij} be the nearest harbor number at each grid point (x_i, y_i). The values S_{ij}'s specify the Voronoi regions of the boat-sail Voronoi diagram.

Algorithm 1 (Boat-Sail Voronoi Diagram)

Input: flow function $f(x, y)$ in Ω and the n harbors $q_1, q_2, \cdots, q_n$.
Output: Arrival time T_{ij} and the nearest harbor number S_{ij} at each grid point.
Procedure:

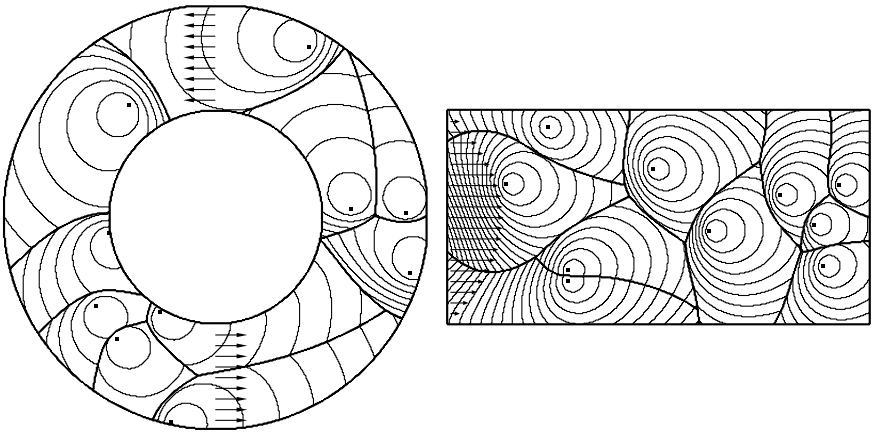
1. For $k = 1, 2, \cdots, n$, set $T_{ij} \leftarrow 0$ and $S_{ij} \leftarrow k$ for harbor q_k, and $T_{ij} \leftarrow \infty$ for all the other grid points.
2. Name the grid points $q_1, q_2, \cdots, q_n$ as "frontier", and all the other grid points as "far".
3. choose the "frontier" point $p = (x_i, y_i)$ with the smallest value of T_{ij}, and rename it as "known".
4. For all the neighbors of p that are not "known", do 4.1, 4.2 and 4.3.
   4.1 If p is "far", rename it as "frontier".
   4.2 Recompute the value of T_{ij} by solving the equation (12) together with the condition (13).
   4.3 If the recomputed value T_{ij} is smaller than the current value, update T_{ij} and also update S_{ij} as the harbor number of the neighbor grid points whose values are used in solving the equation (12).
5. If all the grid points are "known", stop. Otherwise go to Step 3.

Let N be the number of the grid points in Ω. Then, we can prove that Algorithm 1 runs in $O(N \log N)$ time; see Sethian [11] for the derivation of this time complexity.

5 Numerical Examples

5.1 Voronoi Diagram

We show two examples of the Voronoi diagrams in the flow field computed by Algorithm 1. Here, we assume that the speed F of a boat be 1. The arrows in the



(a) (b)

Fig. 5. Voronoi diagrams in the flow fields.

figures represent the directions and the relative speeds of the flow in the field. The thin curves express the isoplethic curves of the first arrival time, and the thick curves express the boundaries of Voronoi regions.

The first example (Fig. 5(a)) is the Voronoi diagram in the circular flow $f = (-0.7 \sin\theta, 0.7 \cos\theta)$ in a doughnut region $\{(x, y) \mid 0.25 < x^2+y^2 < 1\}$ generated by 10 generators. The second example (Fig. 5(b)) is the Voronoi diagram in the flow field $f = (0.7(1 - y^2), 0)$ in a rectangular region $\{(x, y) \mid -1 < y < 1\}$ generated by 10 generators.

5.2 Simulation to Forest Fires

The forest fire is one of the natural phenomena which sometimes happen in the world. It is important for us to estimate how fire extends. In order to predict it, we may have to know a variety of natural conditions. However, it is difficult to take account of all conditions. Hence, we consider the direction and the strength of the wind and the speed at which fire spreads, and simulate the forest fire.

We assume the following. If there is no wind, the extension of fire is isotropic. If the wind blows, on the other hand, the fire can extend faster in the same direction as the wind, while it extends only slowly in the opposite direction. We also assume that forests do not burn so easily as weeds on the plain. Then, by replacing the water flow f with the wind and replacing the maximum speed F of the boat with the speed at which fire spreads out, we can apply our partial differential equation to the simulation of forest fires. Moreover, letting the number of harbor be one and omitting S_{ij} in Algorithm 1, we obtain the algorithm for the simulation.

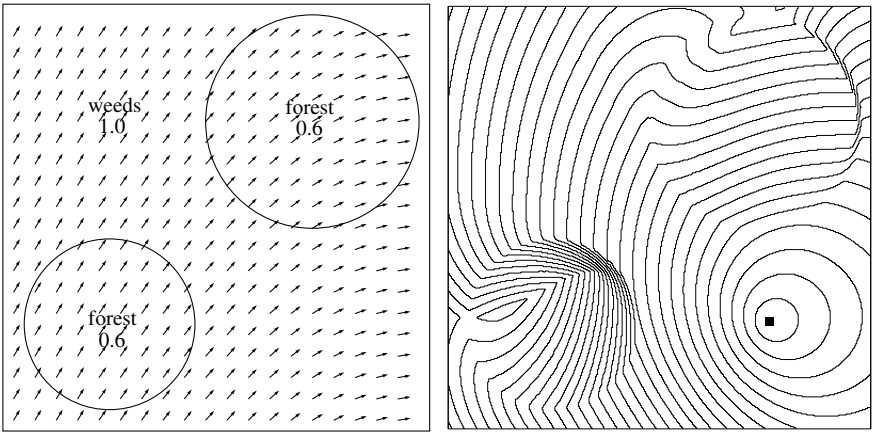


Fig. 6. An Example of Simulation of the fire forest.

Fig. 6(a) illustrates positions of two forests and the direction of the wind. Here, let direction of the wind be $\frac{1}{\sqrt{1+(\pi/2)^2\cos^2(\pi x/2)}}(1, \frac{\pi}{2}\cos(\frac{\pi x}{2}))$ in the square region $\{(x, y) \mid 0 < x < 1, 0 < y < 1\}$. Suppose that the inside of circles are the forests and that the extension speed of the fire in the forests is 0.6 and the speed at other place is 1.0. Fig. 6(b) shows the result of the simulation of the forest fire. A square point represents the initial ignition and curves show the frontiers of the fire spreading out from the initial point time by time.

6 Concluding Remarks

We introduced the boat-sail distance and the associated Voronoi diagram. In order to compute this distance and its Voronoi diagram, we derived a partial differential equation satisfied by the first arrival time, constructed a new stable scheme for solving this equation by extending the first marching method, and showed computational examples of our method.

The concept of the boat-sail distance is natural and intuitive, but the computation is not trivial. Actually the definition of the boat-sail distance given by the equations (2) and (3) does not imply any explicit idea for computing this distance, because the shortest path is unknown. The first reason why we can obtain the efficient computation for the boat-sail distance is that we concentrated on the first arrival time as the unknown function and derived an partial differential equation without time variable. Since the obtained equation is quadratic, we could use the same idea as the fast marching method.

Moreover, by replacing the water flow with the wind and replacing the maximum speed of the boat with the speed at which fire spreads out, we simulated the forest fire.

One of our future work is to construct the second order FEM-like scheme working on irregular triangle meshes. The scheme derived in this paper works on grid meshes. However, if boundary forms become complex, the second order FEM-like scheme on irregular triangle mesh is needed in order to compute the equation more accurately and more stably.

Acknowledgment. This work is supported by the 21st Century COE Program of the Information Science and Technology Strategic Core, and the Grant-in-aid for Scientific Research (S)15100001 of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. B. Aronov: On the geodesic Voronoi diagram of point sites in a simple polygon. *Algorithmica*, vol. 4 (1989), pp. 109–140.
2. P. F. Ash and E. D. Bolker: Generalized Dirichlet tessellations. *Geometriae Dedicata*, vol. 20 (1986), pp. 209–243.
3. R. Courant and D. Hilbert: *Methods of Mathematical Physics Volume II*, Wiley, 1989.
4. S. Fortune: Voronoi diagrams and Delaunay triangulations. In D.-Z. Du and F. K. Hwang (eds.): *Computing in Euclidean Geometry*, World Scientific Publishing, Singapore, 1992, pp. 193–233.
5. K. Kobayashi and K. Sugihara: Crystal Voronoi diagram and its applications. *Future Generation Computer System*, vol. 18 (2002), pp. 681–692.
6. D.-T. Lee: Two-dimensional Voronoi diagrams in the L_p-metric. *Journal of the ACM*, vol. 27 (1980), pp. 604–618.
7. T. Nishida and K. Sugihara: Voronoi diagram in the flow field. *Algorithms and Computation, 14th International Symposium, ISAAC 2003*, Kyoto, Springer, 2003, pp. 26–35.
8. T. Nishida and K. Sugihara: FEM-like Fast Marching Method for the Computation of the Boat-Sail Distance and the Associated Voronoi Diagram. Technical Reports, METR 2003-45, Department of Mathematical Informatics, the University of Tokyo, 2003 (available at http://www.keisu.t.u-tokyo.ac.jp/Research/techrep.0.html).
9. A. Okabe, B. Boots, K. Sugihara and S. N. Chiu: *Spatial Tessellations — Concepts and Applications of Voronoi Diagrams, Second Edition*. John Wiley and Sons, Chichester, 2000.
10. J. A. Sethian: Fast marching method. *SIAM Review*, vol. 41 (1999), pp. 199–235.
11. J. A. Sethian: *Level Set Methods and Fast Marching Methods, Second Edition*. Cambridge University Press, Cambridge, 1999.
12. K. Sugihara: Voronoi diagrams in a river. *International Journal of Computational Geometry and Applications*, vol. 2 (1992), pp. 29–48.

Incremental Adaptive Loop Subdivision

Hamid-Reza Pakdel and Faramarz Samavati

University of Calgary, Calgary, Canada
{hrpakdel,samavati}@cpsc.ucalgary.ca

Abstract. In this paper, a new adaptive Loop subdivision algorithm is introduced. Adaptive subdivision refines specific areas of a model according to user or application needs. Our algorithm extends the specified area such that when it is adaptively subdivided, it produces a smooth surface with visually pleasing connectivity. As adaptive subdivision is repeated, subdivision depth changes gradually from one area of the surface to another area. This smooth transition is analogous to anti-aliasing.

1 Introduction

A subdivision algorithm defines a smooth curve or surface as the limit of a sequence of successive refinements [10]. Subdivision surfaces were first introduced by Catmull and Clark [4] and Doo and Sabin [2] in 1978 as extensions to curve subdivision algorithms. These surfaces are suitable for creating smooth models and are widely used in the entertainment industry [3]. For the purpose of this research, we have focused on Loop subdivision [7] as it has a simple approximating subdivision rule, has local support and produces $\mathbf{C}^2$ surfaces except at extraordinary vertices where the surface is $\mathbf{C}^1$.

Traditionally, subdivision algorithms are applied to the whole input model. For example, in Loop subdivision each face of the input mesh is divided into four. A model with approximately 1000 faces will have about 16000 faces after only two subdivision steps. Occasionally there is no need for a model to be smooth or detailed in all areas. For example, subdivision of a flat surface still yields a flat surface, or subdividing triangles smaller than the pixel size of the screen does not add to the visual quality of the model. Other examples include artistic drawing and smooth silhouette generation. Lastly, users may need a detailed view of portions of the mesh independent of any factors related to the geometry of the model. In these cases, adaptive subdivision can produce an optimal mesh according to specific application needs by subdividing only certain areas of the input mesh.

Adaptive subdivision can be categorized into two subproblems. First, a selection area for subdivision must be defined [1,5,8,11]. Secondly, the mesh must be retriangulated to remove cracks that are caused by a difference in subdivision depth of adjacent faces [1,9,11] because these cracks prevent proper rendering and processing of the surface.

Our research in this paper addresses the second subproblem of adaptive subdivision. One algorithm proposed by Amresh, Farin and Razdan cuts a triangle into two, three, or four triangles depending on the number of cracks the face has [1]. Another similar algorithm, called red-green triangulation [9], cuts the faces into two if the face has only one crack, otherwise it cuts the face into four.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 237–246, 2004.
© Springer-Verlag Berlin Heidelberg 2004

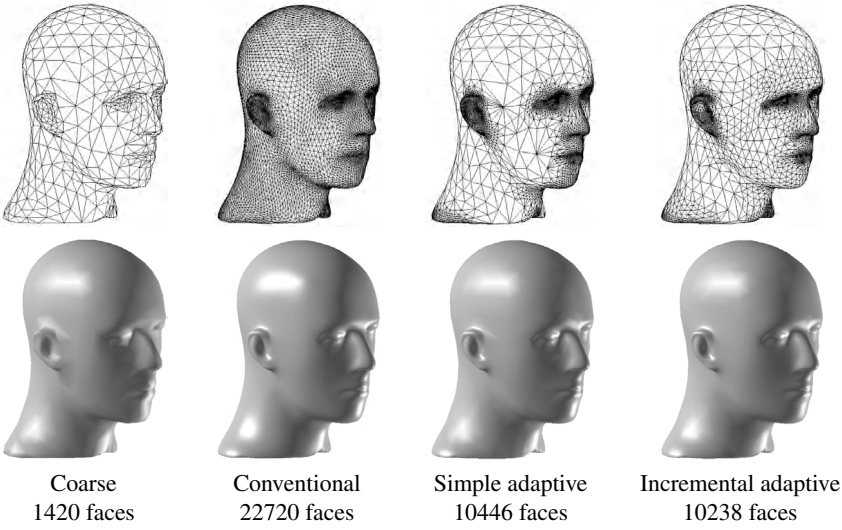


Fig. 1. Comparison of conventional Loop subdivision, simple adaptive subdivision and our incremental adaptive algorithm. In the adaptive cases, only high curvature areas of the mesh are subdivided.

Our contribution in this paper is a new adaptive subdivision algorithm that ensures neighboring faces differ by no more than one subdivision depth. Our algorithm allows vertices within the subdivision area to maintain their connectivity similar to when subdivision is applied to the whole mesh. With our approach, if a specific area is repeatedly subdivided, there is no abrupt change in the subdivision depth. In additional, newly added edges that remove cracks are spread across the mesh. The result is that the produced subdivision surface is smooth, with a progressively changing subdivision depth. Figure 1 shows the result of our adaptive Loop subdivision compared to conventional Loop subdivision and simple adaptive subdivision. We shall call this method *incremental adaptive subdivision*.

Section 2 gives an overview of Loop subdivision, discusses adaptive subdivision algorithms and some of their drawbacks. Incremental adaptive Loop subdivision, our contribution, is presented in Sect. 3 along with a discussion on how it addresses the issues of other adaptive subdivision algorithms. Results and applications of our algorithm, in particular interactive modeling of subdivision surfaces are shown in Sect. 4.

2 Background

2.1 Loop Subdivision

Figure 2 shows three levels of Loop subdivision. At each level, the input mesh is converted to a finer mesh by a simple quadrisection operation followed by averaging of vertices that guarantees a smooth limit surface. Figure 3 shows these averaging rules for the existing (even) and new (odd) vertices.

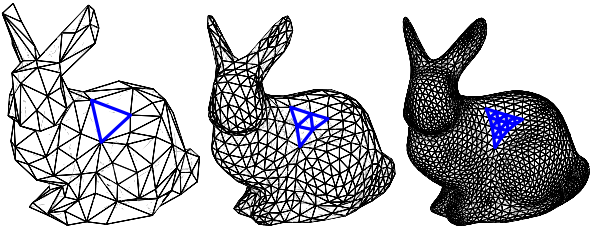


Fig. 2. From left to right: three levels of Loop subdivision.

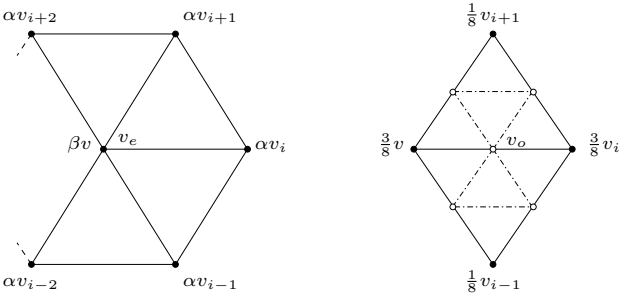


Fig. 3. Coordinate of even (v_e) and odd (v_o) vertices. $\alpha = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}cos(\frac{2\pi}{n})\right)^2\right)$ and $\beta = 1 - n\alpha$ for vertex with valence n. • denotes existing and ∘ denotes newly inserted vertex.

At the limit, the Loop subdivision surface is $\mathbf{C}^2$ everywhere except at extraordinary vertices where it is $\mathbf{C}^1$. In a triangular mesh, vertices with valence of six are ordinary, otherwise extraordinary. As Fig. 3 shows, even vertices from subdivision operation keep their valence while odd vertices are always ordinary. Since odd vertices are created on the edges, at the limit, most vertices of the subdivision surface are ordinary and the surface is $\mathbf{C}^2$.

2.2 Adaptive Loop Subdivision

In adaptive Loop subdivision only a subset of the triangles of the input mesh are subdivided. Selecting which area to subdivide depends on application specific factors. It can be either user-defined or selected based upon specific criteria. Adaptive subdivision produces cracks between triangles that have different subdivision depth. These cracks must be removed if the mesh needs to be further edited, processed or subdivided. This section reviews existing crack removal algorithms and outlines some of the issues that our method is able to address.

Selection Criteria. Users may choose to subdivide only a specific region of the mesh. As we show in Sect. 4, an area of the model may not have the required detail for precise

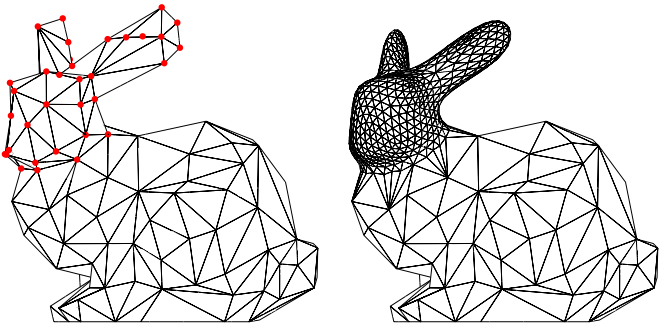


Fig. 4. Adaptive subdivision of user defined area.

modeling. Artists may want to emphasize part of a scene by increasing the detail of that area. In these cases, users can either select vertices or triangles of the model to subdivide. If vertices are selected, any triangle that has two or more selected vertices is subdivided. Figure 4 shows Loop subdivision of user defined area of the bunny model.

Surface curvature is another selection criteria for adaptive subdivision. In Fig. 5, Gaussian curvature [8] of each vertex, computed from its sums of Voronoi area, is used to refine high curvature areas of the model, as these areas generally need more approximation. Dihedral angle, the angle between normals of adjacent faces, can also be used as an approximation of surface curvature. Albeit dihedral angle is not as accurate as Gaussian curvature, it is more efficient to compute and it is still an important determinant of the surface curvature. Another selection criteria can be closeness of the surface to the limit subdivision surface. It can be evaluated by using limit subdivision masks that are obtained from eigenanalysis of subdivision operation. The further the surface from its limit, the more it must be subdivided.

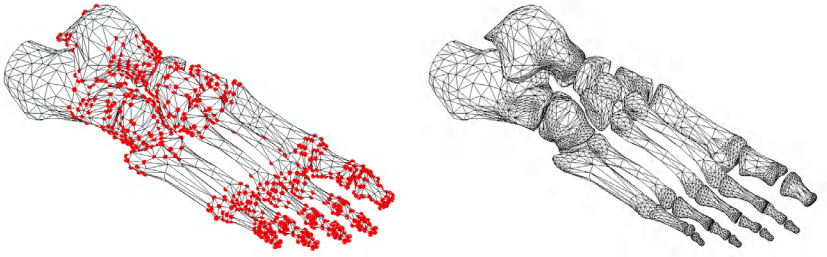


Fig. 5. Adaptive subdivision of the foot bones. The selected vertices indicate higher curvature ares of the surface.

In [6], a general approach to the selection criteria for adaptive subdivision is taken. The selection area is defined as a set of faces that satisfy some Degree of Interest (DoI) function which may or may not be based on the geometric properties of the model. For example, to generate smooth silhouettes, the DoI can be set to take the normal of faces into consideration and subdivide faces that share edges on the silhouette boundary.

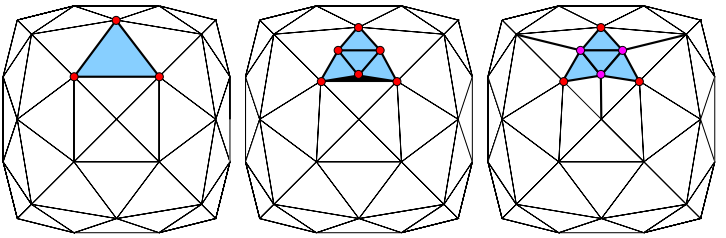


Fig. 6. A triangle of the mesh is selected for subdivision in the left picture. In the middle, cracks are created between faces which differ in subdivision depth. On the right image, cracks are removed by bisecting faces with lower subdivision depth.

Removing Cracks. Figure 6 shows a case where only one triangle of a mesh is subdivided. Neighboring faces with different subdivision depth create cracks in the mesh. This is because the shared edge between these faces contains a vertex with incomplete connectivity. The resulting cracks must be removed so the surface can be further edited, processed or subdivided.

One method of removing cracks is to bisect the face that has not already been subdivided, effectively connecting the vertex with the incomplete structure to its opposing vertex [1]. As shown in Fig. 6, this method introduces T-vertices into the mesh where the face is bisected.

Another method, called red-green triangulation [9], bisects faces that have an edge with one crack, but quadrisects them when there are two or more cracks. Note that cracks may not only be created by neighboring faces, but also by their children from subdivision. Figure 7 shows an example of red-green triangulation. When triangle ABC is subdivided, triangle BEC must be bisected to remove a crack due to edge BC. However, if triangle CEF is also subdivided, then triangle BEC will have two cracks, due to edges BC and CE, and must be quadrisected. To do this, triangle BEC must be reconstructed from triangles BET and ECT and divided into four using the Loop subdivision operation. This process creates a crack due to edge BE and is removed by bisecting triangle BDE. Care must be taken when a chain bisection and quadrisections are performed as some faces may be bisected and quadrisected at the same subdivision depth.

In [11], Zorin, Shröder and Sweldends use the concept of a restricted mesh for adaptive subdivision. In this algorithm, the mesh is stored in a tree data-structure and the leaf nodes of the tree represent the last subdivision depth. Rather than handling cracks right after subdivision, they are removed during the rendering stage of the algorithm. Before subdivision, face cracks are removed by refining the parents of neighboring faces until all vertices have a complete neighborhood. This process ensures that the proper averaging rules are applied during subdivision. After the subdivision, faces that are not needed are discarded. During rendering, cracks are removed in the same manner as [1].

Our algorithm extends the method introduced in [1] to remove cracks after adaptive subdivision. It selects a larger subdivision area than the specified one to maintain a restricted mesh [11] during subdivision. We will now discuss some of the drawbacks of simple bisection to remove cracks and show in the next section how our algorithm addresses these issues.

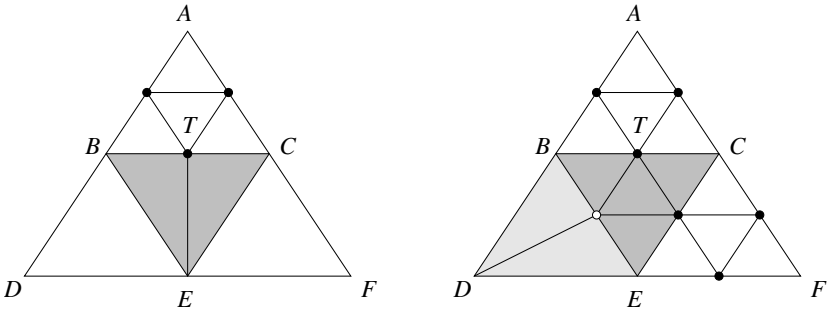


Fig. 7. Red-green triangulation. Left diagram: After subdividing $\triangle ABC$,$\triangle BEC$ must be bisected. Right diagram: If $\triangle CEF$ is subdivided, then $\triangle BEC$ would have two cracks and must be quadrisected resulting in bisection of $\triangle BDE$.

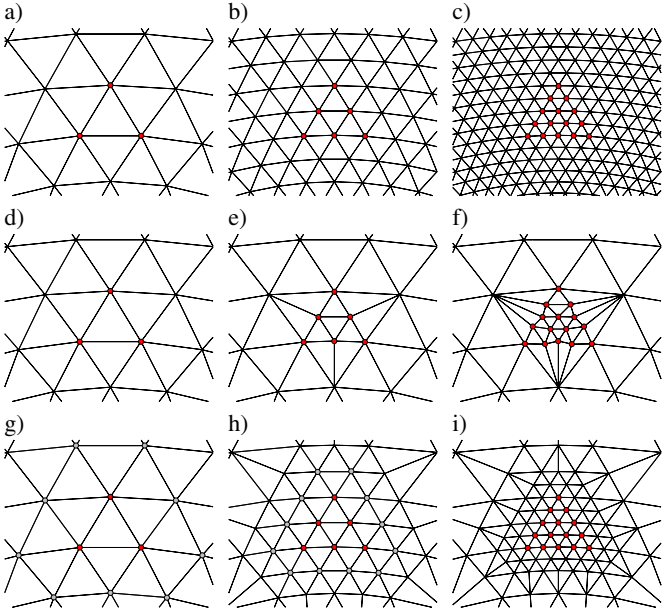


Fig. 8. Comparison of conventional Loop subdivision to adaptive subdivision presented in [1] and our incremental algorithm. The dots indicate selected vertices for subdivision.

Repeated Subdivision. Bisecting faces to remove cracks as presented in [1] has three main problems:

1. T-vertices are always extraordinary. If a selected area is subdivided repeatedly, then the geometry of T-vertices will vary from when subdivision is performed on the whole input mesh. This effect can clearly be seen in Fig. 8f when compared to Fig. 8c.
2. Figure 9 shows how repeated subdivision of a selected area generates high valence vertices as faces are bisected to remove cracks from the mesh. High valence ver-

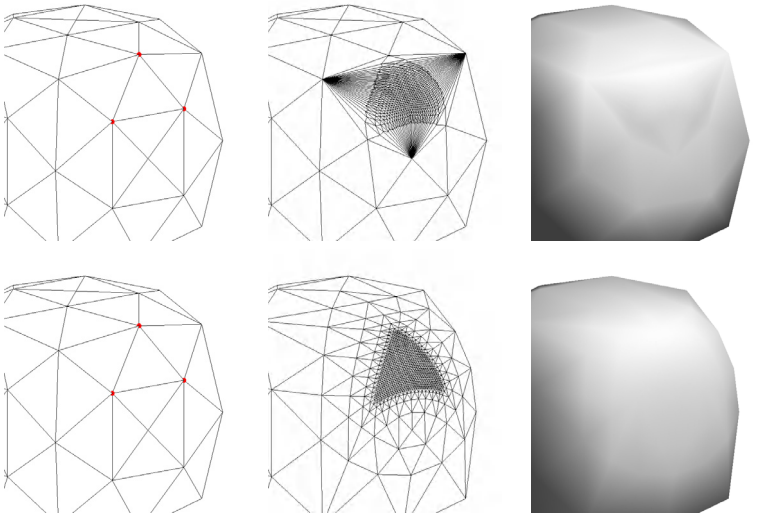


Fig. 9. Top row: Repeated simple adaptive subdivision results in high valence vertices, abrupt change of subdivision depth and incorrect geometry of the surface. Bottom row: Repeated incremental adaptive subdivision generates a smooth surface that progressively changes in subdivision depth.

tices create long and skinny faces which are generally undesired in modeling and rendering applications.

3. After a number of subdivision steps, there is a large difference in subdivision depth between neighbouring triangles. This causes abrupt change of connectivity and curvature across the surface. The bump created around the selection area in Fig. 9c is due to this effect.

3 Incremental Adaptive Loop Subdivision

To overcome the disadvantages of the algorithm presented in [1] a restricted mesh [11] is needed because it ensures that odd vertices remain ordinary during subdivision. Red-green triangulation removes vertices with a large valence and avoids abrupt changes of connectivity by quadrisecting faces with two or more cracks, but it still suffers from the different geometry problem that is discussed in the previous section, unless it also maintains a restricted mesh during subdivision.

Figures 8g, 8h and 8i outline our incremental adaptive subdivision algorithm. In the general case where no boundaries exists, the faces that are immediately outside the selection area are included in the subdivision process. As shown in Fig. 8h, if the selected area is subdivided again, then the faces immediately to this area are tagged as well as subdivided. In practice, rather than tagging faces for subdivision, vertices are tagged as either `selected` or `progressive`. Before subdivision, all vertices within the selection area are enumerated and if any of their neighboring vertices is not selected, then it is

tagged as `progressive`. Faces with two or more tagged vertices are subdivided while the rest remain untouched. Note that boundary cases are automatically handled by this algorithm.

In contrast to [1], our algorithm removes cracks *outside* the selection area so they no longer affect the selection region of the mesh which is important to the user or application. Odd vertices within the selection area remain ordinary because the faces within this area and the immediate neighbors outside it have the same subdivision depth. Another result of our algorithm is that faces are at most one subdivision depth apart which has two consequences. First, the connectivity of faces does not change abruptly. The closer the faces are to the area that is repeatedly subdivided, the higher their subdivision depth. Sudden changes in subdivision depth are analogous to aliasing in rendering. Our algorithm applies an anti-aliasing method to create smooth transition of subdivision depth from one area of the mesh to another. Second, faces rarely have more than one crack, so high valence vertices and long triangles are avoided. Figure 9 compares the results of repeatedly subdividing a triangle of a coarse mesh using the adaptive and incremental adaptive methods.

4 Results and Applications

In practice, our incremental method creates subdivision surfaces that are smooth and progressively change in subdivision depth. Removing cracks by adding edges creates a number of extraordinary vertices in the mesh. At the limit, subdivision surfaces are $\mathbf{C}^1$ at the extraordinary vertex. Hence, they are undesirable, but also unavoidable in all adaptive subdivision algorithms. While our algorithm attempts to create these vertices outside the selection area, allowing for $\mathbf{C}^2$ surface in the case of regular meshes, they still affect the final surface if the model is globally subdivided. Since we limit the subdivision depth to at most one and spread the edges across the mesh, the surface is minimally affected by them.

We have developed a subdivision surface editor based on our algorithm. It allows users to interactively subdivide portions of a model using different selection methods. For example, users may select a percentage of high curvature areas of the model to be subdivided, or manually select an area of the model. In Fig. 10, parts of a gear model are selected by the user and incrementally subdivided. Fig. 11 shows a subdivision "pen" that allows users to interactively refine the model by drawing on it. As the pen moves over the surface, the faces are incrementally subdivided. The slower the pen moves, the more the area underneath is refined to reflect more details on the surface.

5 Conclusion

Conventional subdivision is useful in creating surfaces that are overall smooth and detailed. Adaptive subdivision allows creating surfaces with different subdivision depths tailored for specific applications. Existing adaptive subdivision algorithms produce surfaces that have either improper geometry or bizarre connectivity. In this paper, we have introduced a new adaptive subdivision algorithm that additionally subdivides the closest faces around the selection area, in effect creating a surface that gradually increases in

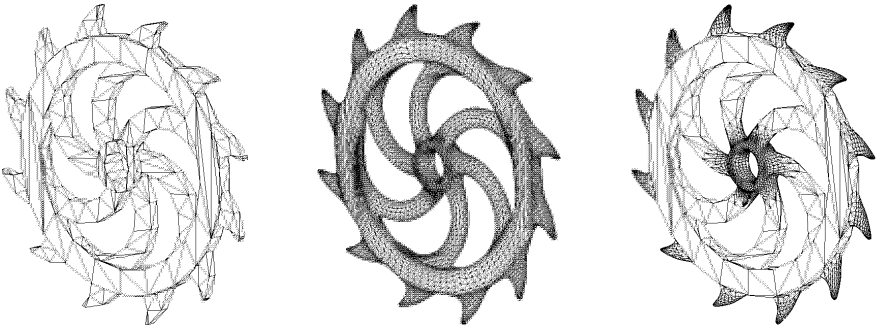


Fig. 10. Using incremental subdivision to smoothen specific areas of a the gear. Left image: original mesh. Center image: after two Loop subdivision steps. Right image: incremental subdivision of the spikes and the center of the gear.

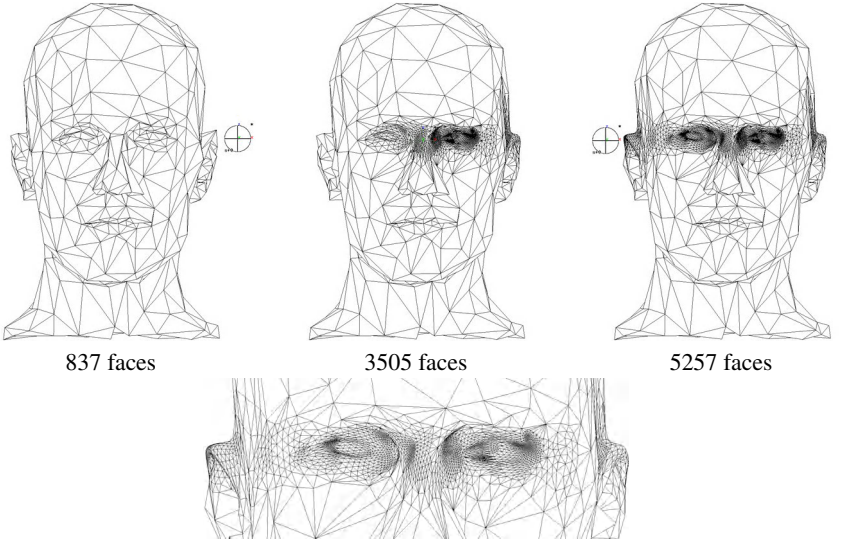


Fig. 11. Pen based real-time incremental subdivision. Top row: From left to right, as the pen moves over the ears and eyes of the figure head, the faces are incrementally subdivided. Note that this model would have 13392 faces after only two subdivision steps applied to the whole mesh. Bottom row: Zoomed in on the eyes.

subdivision depth. The smooth surfaces created by our algorithm have proper connectivity and geometry. Incremental adaptive Loop algorithm can be effectively used in a number of applications, including modeling and finite-element analysis.

Acknowledgement. We would like to thank Colin Smith and Peter MacMurchy for their helpful comments and suggestions. This work is partially supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

1. A. Amresh, G. Farin, and A. Razdan. Adaptive subdivision schemes for triangular meshes. In G. Farin, H. Hagen, and B. Hamann, editors, *Hierarchical and Geometric Methods in Scientific Visualization*, pages 319–327, 2003.
2. E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, September 1978.
3. T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. *Computer Graphics*, 32(Annual Conference Series):85–94, August 1998.
4. D. Doo and M. Sabin. Behavior of recursive subdivision surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, September 1978.
5. N. Dyn, K. Hormann, S. J. Kim, and D. Levin. Optimizing 3D triangulations using discrete curvature analysis. In *Mathematical Methods for Curves and Surfaces: Oslo 2000*. Vanderbilt University Press, 2001.
6. T. Isenberg, K. Hartmann, and H. König. Interest value driven adaptive subdivision. In T. Schulze, S. Schlechtweg, and V. Hinz, editors, *Simulation und Visualisierung*, pages 139–149. SCS European Publishing House, March 2003.
7. C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, August 1987.
8. M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds *VisMath*, 2002.
9. S. Seeger, K. Hormann, G. Häusler, and G. Greiner. A sub-atomic subdivision approach. In T. Ertl, B. Girod, G. Greiner, H. Niemann, and H. P. Seidel, editors, *Proceedings of the Vision Modeling and Visualization Conference 2001 (VMV-01)*, pages 77–86, Berlin, November 2001. Aka GmbH.
10. D. Zorin and P. Schröder. Subdivision for modeling and animation. Course notes of Siggraph, 2000.
11. D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. *Computer Graphics*, 31(Annual Conference Series):259–268, August 1997.

Reverse Subdivision Multiresolution for Polygonal Silhouette Error Correction

Kevin Foster, Mario Costa Sousa, Faramarz F. Samavati, and Brian Wyvill

Department of Computer Science
University of Calgary,
Calgary, Canada
{fosterk,mario,samavati,blob}@cpsc.ucalgary.ca

Abstract. This paper presents a method for automatic removal of artifacts that appear in silhouettes extracted from polygonal meshes due to the discrete nature of meshes and numerical instabilities. The approach works in object space on curves made by chaining silhouette edges and uses multiresolution techniques based on a reverse subdivision method. These artifact-free curves are then rendered in object-space as weighted 3D triangle-ribbon strips.

1 Introduction

There has been significant research in non-photorealistic rendering focusing on quality silhouette extraction and rendering, in particular for 3D mesh-based silhouette line stylization algorithms [8,11,12,15]. Such algorithms are usually organized in four main steps: (1) extraction of individual silhouette edges from the mesh; (2) linkage of silhouette edges together to form long, connected paths, or chains; (3) removal of silhouette artifacts from the chains; (4) stylization of the strokes which involves two main sub-processes: smoothing the chain by fitting splines or using an interpolation/approximation scheme and controlling line quality attributes along the chain such as width and brightness.

A problem with extracting silhouette curves from polygon meshes is that the resulting curves may contain jagged artifacts because of numerical instability and unsuitable edges from the mesh (the mesh is a discrete approximation of a surface). These artifacts compromise the quality of the stroke stylization process and subsequent rendering results. Although there is a great deal of work which extracts silhouettes from polygonal meshes, there are few examples that attempt to correct errors and artifacts that can be created when this extraction takes place (*step 3*).

In this paper, we introduce a new approach to remove artifacts from chains of silhouette edges based on multiresolution. Because silhouettes created from polygonal meshes have a discrete nature, use of multiresolution systems that directly operate on discrete data are fitted effectively. Samavati and Bartels[1,13] provide this kind of multiresolution based on reversing subdivision. In their system, resolution can be increased and decreased efficiently without use of wavelets.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 247–256, 2004.
© Springer-Verlag Berlin Heidelberg 2004

We employ this kind of multiresolution to remove silhouette artifacts automatically and efficiently. Furthermore, we can also use subdivision consistently with the multiresolution filters to contribute to the stroke stylization step.

2 Related Work

(1) Object-space silhouette extraction: There are many methods that extract silhouettes from polygonal meshes, including systems based on probabilistic testing [11], "Gauss Maps" [6], duality [7], cone maps[14] and adjacency information[2]. Any of these methods can be used with our error-removal system, provided they create linked silhouette chains. In this work, we extend the "edge-buffer" method[2] to create these chains.

(2) Removing silhouette artifacts: Works in this area either (1) correct errors from silhouette chains created from the actual mesh edges [8,12] or (2) create new, more suitable, silhouettes without use of the edges in the mesh [3, 7]. Correa et al.[3] avoid errors by creating 2D *u,v-images* which are basically projected images of the 3D scene with special colors on different u,v coordinates on the mesh. Their system analyzes pixel-neighborhoods and creates curves from the areas that contain silhouettes. Mesh edges are not used in this process; thus errors are avoided. Northrup and Markosian[12] remove errors by rendering raw silhouettes to image-space and case-checking. This includes elimination of undesirable silhouettes, redefinition of uneven endpoints so that they correspond and joining of edges to create smooth chains. Isenberg et al.[8] also correct silhouette errors directly from the edges using case-checks and solutions. However, their corrections are preformed in object-space. Hertzmann and Zorin[7] present an object-space approach that avoids errors by creating more suitable silhouette edges. These new edges are created by approximating points on the mesh edges where the silhouette would cross if the mesh was a smooth surface.

Our method, like Hertzmann and Zorin's[7], is general—we remove all errors without requiring classification of errors and evaluation of fixes. However, like Isenberg et al.[8] and Northrup and Markosian[12], our system removes errors from silhouette chains created from edges in the mesh instead of procedurally generating new edges. This approach desirable due to the speed and simplicity of extracting silhouette edges from a mesh.

(3) Multiresolution methods: Finkelstein and Salesin[5] demonstrate the first use of multiresolution in NPR with a curve-editing system based on wavelets. Furthermore, Kirsanov et al. [10] use coarsening methods to simplify silhouettes from detailed polygonal meshes. More information on this type of multiresolution is found in Stollnitz et al.[16]. We use a different type of multiresolution, "*local*" [1] and "*global*" [13] multiresolution, based on reversing subdivision to remove errors and provide a better system to simulate smooth pen strokes.

We now describe the main steps of our algorithm: (1) Create silhouette chains (Sec. 3), (2) Apply our multiresolution system to remove errors (Fig. 3, Sec. 4); and (3) Stylize the chains (Sec. 5). We then present and discuss results (Sec. 6) and provide conclusions and directions for future work (Sec. 7).

3 Silhouette Extraction

The definition of a silhouette edge for object-space methods is any edge shared by one front-facing polygon and one back-facing polygon. Upon loading a mesh, our system constructs an edge-buffer [2]. The edge-buffer, which can be viewed as an indexed graph of edges, is a fairly compact data structure that provides a fast lookup method requiring, for each frame, two binary operations per edge to extract silhouettes in object-space. Further details are supplied in [2].

3.1 Chains

The algorithm proceeds to the point where the edge-buffer has been traversed and all silhouette edges properly extracted. To better reproduce the artistic style described in Sec. 1, we use a two-pass algorithm to create a small number of long silhouette chains. As a first pass, our system links the silhouette edges on the model by finding the connected components of the edge-buffer. As a second pass, our system finds the matching vertex numbers on the bounds of each chain and joins these chains. If more than two chains can be linked, we join any chains that will create a loop first. Looping chains take precedence because our multiresolution system (Sec. 4) handles looping and non-looping chains slightly differently (non-looping chains are interpolated at the ends) and if a chain that should be looping is instead identified as two separate strokes, small artifacts might be created due to the interpolation at the ends of the chain.

This chaining method cannot guarantee the longest connected chains. However, it does generate satisfactory long chains for use with the multiresolution filters.

3.2 Artifacts

The chains extracted in the processes described above may contain artifacts such as zig-zags, overlaps and loops (Fig. 1). Such artifacts exist for two main reasons: (1) numerical instabilities in silhouette edge detection where many of the faces are viewed nearly edge-on; (2) meshes are just approximations of the underlying continuous surfaces and the edges that make them up are almost always unsuitable to be used as silhouette edges.

The set of four images in Fig. 1(b) illustrate different combinations of these artifacts. Silhouettes for these images have been calculated for an angle other than that displayed. Observe the black line, which is the actual silhouette, the unshaded front-facing polygons and the shaded back-facing polygons. As the silhouette crosses the surface, it moves back and forth across some invisible threshold, sometimes by many edges at a time. Clearly, edges taken directly from the mesh are not ideal to construct the silhouette. The invisible threshold that the extracted silhouette crosses is approximately where it should actually appear.

We interpret silhouette artifacts from the point of view of low and high-frequency portions of the silhouette curve. The extracted silhouette can be viewed as high-frequency noise components along the real silhouette curve. The

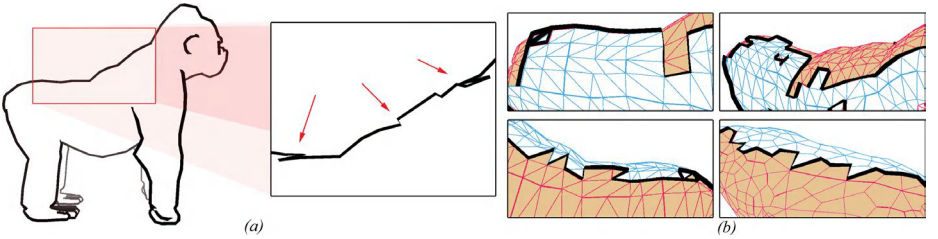


Fig. 1. (a) The silhouette of an ape mesh with highlighted errors. **(b)** Four images showing various silhouettes and underlying mesh that generated them. The silhouettes are presented at a perturbed view to provide a better understanding of the cause of the errors. Shaded polygons are back-facing.

challenge is to remove the high-frequency noise which occurs sporadically along the chain. We meet this challenge by using multiresolution filters, as described in the next section.

4 The Multiresolution Approach

The algorithm proceeds to where complete chains have been constructed from the silhouette edges. We denote these ordered sets of points as C^{k+1}. Using multiresolution, C^{k+1} can be decomposed to a low-resolution approximation C^k and a set of high frequency details D^k. Thus, C^k shows overall sweep of the silhouette and D^k shows waves and zigzags of the silhouette. In functional view, C^{k+1} is coefficient vector of high resolution scaling functions, C^k is coefficient vector of low resolution scaling functions and D^k is coefficient vector of Wavelet functions. The original data C^{k+1} can at any time be reconstructed from C^k and D^k. The process of transforming C^{k+1} to C^k and D^k is called *decomposition* and generating the original data C^{k+1} from C^k and D^k is called *reconstruction*. These can be applied to C^{k+1} more than one time. We can specify the multiresolution operations in term of the banded matrices A^k, B^k, P^k and Q^k. The matrix A^k transforms C^{k+1} to C^k:

$$C^k = AC^{k+1} \tag{1}$$

and B^k extracts details:

$$D^k = BC^{k+1} \tag{2}$$

P and Q act on C^k and D^k to reconstruct C^{k+1}

$$C^{k+1} = PC^k + QD^k \tag{3}$$

These matrices have a regular structure for every resolution. The only difference between A^k and A^{k-1} is their size. Consequently, the superscript of matrices can be removed. Because of the regularity of these matrices, they can viewed as filters that operate on C^{k+1}, C^k and D^k.

In order to find these four matrices, most multiresolution research works in the area of wavelets. In the case of smooth curves, the resulting wavelets are not very interesting (see appendix of Finkelstein[5] or page 94 of Stollnitz et al.[16]). With our method, C^{k+1} is a discrete approximation of a smooth curve and we just need to use appropriate A, B, P and Q and we do not need wavelets explicitly. Therefore, a discrete approach of multiresolution systems that directly operates on discrete data is fitted here more effectively. Bartels and Samavati[1] and Samavati and Bartels[13] provide this kind of multiresolution system based on reversing subdivision. In this kind of multiresolution, decomposition and reconstruction can be done efficiently without use of wavelets. They have also shown their results are more effective for data sets than conventional wavelets. In this work, we use their multiresolution filters that are constructed based on reversing Chaikin subdivision, Cubic B-Spline subdivision and Dyn-Levin subdivision. We present the masks of their Cubic B-Spline subdivision in Fig. 2a. These filters are much easier than their counterparts in Finkelstein and Salesin[5] and Stollnitz et al.[16].

For implementation, we just need to apply A and B on C^{k+1} to obtain C^k and D^k. Again by applying P and Q filters on C^k and D^k, or a modified version of D^k, we can reconstruct C^{k+1}. Note that these processes are simple linear time operations which do not use any extra storage. The resulting filters of Bartels and Samavati[1] are obtained based on solving the best C^k via a local least squares problem while the resulting filters in Samavati and Bartels[13] are obtained based on a global least squares problem. We call these two approaches *local* and *global* multiresolution. Note that these filters produce the optimum solution intrinsically without any extra-work in implementation. In the case of local multiresolution (Fig. 2), the implementation is very simple and straightforward. However, C^k is just a good approximation of C^{k+1} in a local sense. In contrast, C^k found from C^{k+1} with a global manner is the best solution possible (although it is more complicated than the local one). In fact, in the global case, the matrices A and B are full matrices. Nevertheless, they still have the regular structure. In order to achieve linear time operations, we solve the following banded system for decomposition [13]:

$$(P^t P)C^k = P^t C^{k+1} \tag{4}$$

$$(Q^t Q)D^k = Q^t C^{k+1} \tag{5}$$

In our experiments comparing local and the global multiresolution for silhouette error removal, we have found that the global MR generally creates better results (Sec. 6). However, the drawback of this approach is the need of solving the systems in equations 4 and 5.

4.1 Error Removal Pipeline

In this section, we provide details on how these filters can be used to eliminate silhouette artifacts. Our multiresolution pipeline consists of decomposing silhou-

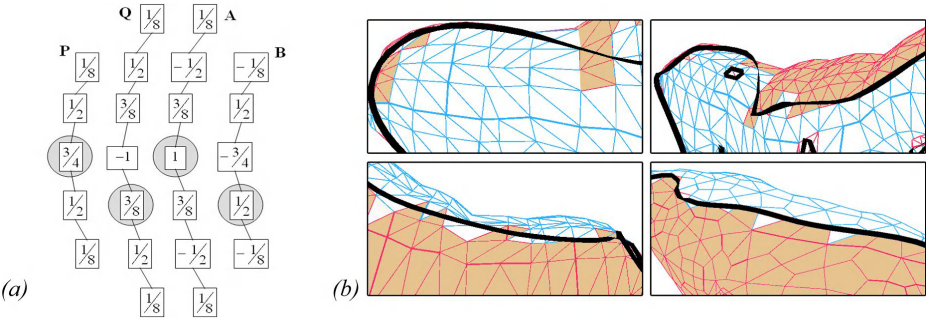


Fig. 2. (a) The bands of the matrices for Cubic B-Spline multiresolution (the A, B, P and Q diagrams represent all non-zero entities of a row for the A and B matrices and of a column for the P and Q matrices). The gray circles show the center entity. **(b)** Results of running our system on the silhouettes in Fig. 1b.

ettes to some level of detail, then reconstructing with only a small percent of the high-frequency details to remove errors (Fig. 3).

We modify equation 3 so that it can lessen the amount of details included in reconstruction:

$$\bar{C}^{k+1} = PC^k + eQD^k \tag{6}$$

where e is a scalar between 0.0 and 1.0 that varies the percentage of the details data added to the coarse data. The higher the value of e included, the greater the percent of the details data is included and the closer the stroke gets to the original data extracted.

Recall that the the low frequency path of the raw silhouette chain is generally correct (Fig. 1). The errors are all high-frequency divergences from this path. Since the high frequency portion of the silhouette chain is extracted and stored in details, a lower value for e eliminates more errors as a lower percent of the high-frequency details are included in the reconstructed strokes. We were able to generate accurate strokes suitable for scientific illustration with values from 0.0 to 0.4 for e, depending on the detail in the original mesh. A discussion of this is provided in Sec. 6.

Note that reconstruction can continue to a higher level of detail than the original chain. This is done by eliminating QD^k in equation 6 and results in an increase in smoothness. This is illustrated in the rightmost image in Fig. 3 (note quality improvement on the ape's head).

In our implementation, the user has control over the amount of times to decompose and reconstruct, the method to do this decomposition and reconstruction (Chaikin, Cubic B-Spline or Dyn-Levin), the scope of the method (local or global) and the amount of details to include in the reconstruction (the e value). Note that low-pass filters do not give this level of control.

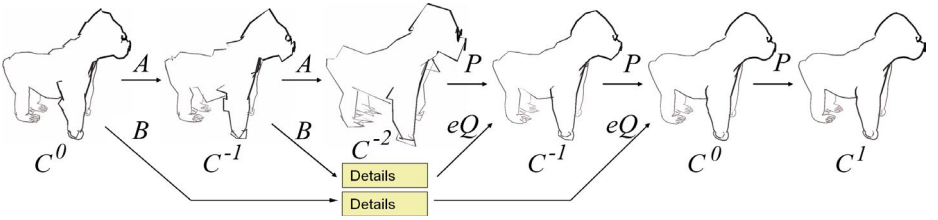


Fig. 3. We use Multiresolution filters to decompose and reconstruct silhouette chains without errors. Here is an example for an ape mesh with 7434 faces. We decompose twice from level C^0 to C^{-2} with **global cubic B-Spline** filters. Then, we reconstruct to level C^0 using minimized details (here, $e = 0.3$). The effect of this process is the removal of errors. Note that we can further process the mesh (to level C^1 or higher) without any details to smooth the strokes. This is equivalent to a subdivision step.

5 Rendering

For the results in this paper, we use the angled-bisector strip method as presented by Northrup and Markosian [12] and vary the weight and intensity of the stroke based on its depth into the scene. To preform Hidden Line Removal (HLR), we rely on the depth buffer. The original mesh is drawn in white and the strokes are drawn, slightly displaced towards the viewer. Thus, any strokes on the back of the surface will be occluded by the white mesh with the z-buffer. This approach does not work well for small meshes because the processed strokes do not follow the exact mesh; however it works well for medium to large size meshes. We leave an exact fast object-space solution to this problem for future work.

6 Results and Discussion

Our system achieves fast computation rates including preprocessing (building the edge-buffer) and rendering (chaining, multiresolution filtering, and stroke stylization). Furthermore, we have found that our method removes most errors with two levels of decomposition and reconstruction and a small value for e. Our method gains speed over other silhouette error correction methods because we do not need to identify errors to remove them. Thus, we do not need a large set of error condition/correction cases that must be evaluated locally for individual portions in the silhouette chain. However, this means that our method can inadvertently remove important features. Our system presents a tradeoff between feature-preservation and quality of filtering (directly related to the value e). Although this is not an issue for detailed meshes (features are preserved even with low levels of e), it can sometimes be impossible to remove errors from silhouettes of simple meshes without losing stroke accuracy (Fig.6).

We now present running times for different mesh sizes and the speed difference between local and global filters. Then we discuss quality of the results with notes on mesh size, user input and the different filter types.

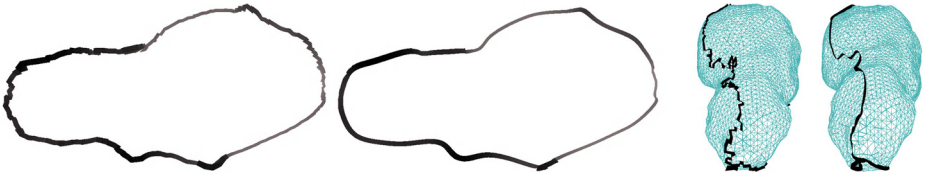


Fig. 4. From left to right: Original silhouettes from an asteroid, the results of processing and alternate views of the strokes with the mesh.

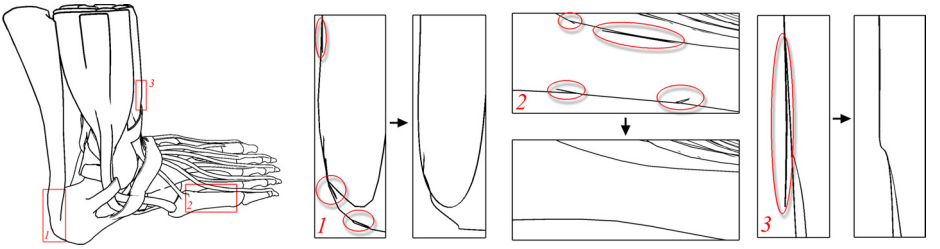


Fig. 5. Removing silhouette errors on large meshes is more important when zooming in on the mesh. Here, we circle the errors on three enlarged areas on the foot and provide our corrected strokes. Note that the errors are removed and the strokes are still very accurate to the mesh.

(1. Timing:) We have found that the local multiresolution filters generate realtime results for medium sized meshes (around 30,000 faces) and interactive rates for larger meshes. With two levels of decomposition and reconstruction and local Cubic B-Spline filters, the ox takes 0.414 milliseconds to filter (Fig.6), the ape 0.825 ms (Figs. 2, 3), the asteroid 1.065 ms (Fig. 4) and the foot 63.887 ms (Fig. 5). These results are ordered in increasing mesh size and are averaged from 256 tests with chains taken from the mesh at different angles. Clearly, our filters are efficient and even large meshes such as the foot run interactively.

As expected, the global multiresolution method is slower. For the ape and asteroid models, two levels of decomposition with global Cubic B-Spline filters take 7.779 and 19.75 ms respectively. This is a large increase over local times, but the method still preforms quickly for less detailed meshes where accuracy is most important. The added accuracy of global methods over local methods is not required for high resolution meshes. Running times and result images were gathered from a 2.65 GHz Pentium 4 with OpenGL/ATI Quadro graphics.

(2. User Input:) We found that medium to large meshes require little or no user-input (Figs. 4, 5). Error free strokes with no accuracy loss can almost always be generated with local multiresolution using two levels of decomposition and reconstruction and some small e value for details. The more detailed the mesh, the smaller e can be while still maintaining accurate strokes. We generally employed $e <= 0.1$ for meshes larger than 10,000 triangles. For smaller meshes (Figs. 3, 6) or for features on larger meshes only defined by several triangles, the user must use a global method (see next point on multiresolution type) and

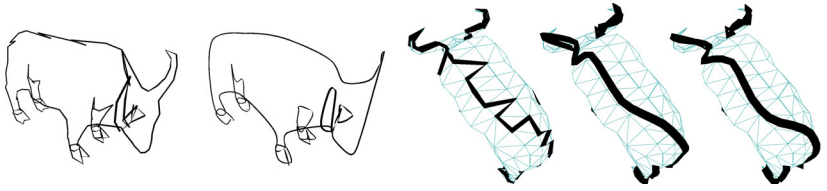


Fig. 6. Left to Right: A silhouette from a low resolution ox mesh, processed strokes with global Cubic B-Spline filters, an alternate view of the original and processed strokes, and processed strokes with global Chaikin filters. Note that the corrected strokes do not adhere well to the original mesh.

carefully adjust the amount of details and the decomposition and reconstruction steps to generate accurate strokes. It is in these situations that varying e results in a noticeable tradeoff between error-removal and feature preservation.

(3. Multiresolution Type:) We have tested the Cubic B-Spline, Dyn-Levin and Chaikin systems [1,13] with local and global multiresolution methods. The global method has produced more accurate results. This increase in accuracy can be seen in the right column of Fig. 2b where global methods are used compared to the left column where local methods are used). However, as presented in the timings section, the expense of global methods rises with mesh size. Fortunately, the extra accuracy given is usually only useful for small to medium sized meshes where the global method preforms in in realtime. For the figures in this paper, we have employed global methods for the smaller meshes in Figs. 2b(right column), 3 and 6 and have employed local methods for the larger meshes in Figs. 2b(left column), 4 and 5.

7 Conclusions and Future Work

We have presented a method to eliminate errors in polygonal silhouettes using multiresolution filters. Our method represents an improvement over previous works because it does not require specialized error/solution cases to remove errors—our solution is general. This improves efficiency over other methods because time is not spent identifying errors and looking up solutions. Furthermore, our system contributes to the stroke stylization step by automatically smoothing coarse chains. Finally, our system complements systems which create coarse approximations of silhouettes from very detailed meshes[10].

The drawback to our method is that it is not exact. We do not guarantee that errors will be removed and that accurate strokes can be generated for all meshes. Our approach can only be used to automatically generate accurate strokes for medium to large size meshes while coarser meshes can require a great deal of user input to create good output without losing detail. Finally, our method does not present a good way to eliminate unessential silhouette chains and an accurate hidden-line removal method must be developed for this system.

Despite these limitations, our method provides a new avenue to remove silhouette errors from polygonal silhouettes that is accelerated and more general. A future extension could be to improve the numerical stability during silhouette extraction with techniques that compute better normal vectors [17]. Furthermore, our system could be used to process non-silhouette strokes as an improvement to traditional B-spline or low-pass filtering methods [4,15]. Finally, our method could be combined with the approach presented by Kalnins et al.[9] for coherent silhouettes.

References

1. Bartels RH, Samavati FF (2000) Multiresolution curve and surface representation by reversing subdivision rules. Computer Graphics Forum, Vol. 18, No. 2: 97–120
2. Buchanan JW, Sousa MC (2000) The edge buffer: A data structure for easy silhouette rendering. Proc. of NPAR'00: 39–42
3. Correa WT, Jensen RJ, Thayer CE, Finkelstein A (1998) Texture mapping for cel animation. Proc. of SIGGRAPH'98:435–446
4. DeCarlo D, Finkelstein A, Rusinkiewicz S, Santella A (2003) Suggestive contours for conveying shape. Proc. of SIGGRAPH'03: 848–855
5. Finkelstein A, Salesin DH (1994) Multiresolution curves. Proc. of SIGGRAPH'94: 261–268
6. Gooch B, Sloan PJ, Gooch A, Shirley P, Riesenfeld R (1999) Interactive technical illustration. 1999 ACM Symposium on Interactive 3D Graphics: 31–38
7. Hertzmann A, Zorin D (2000) Illustrating smooth surfaces. Proc. of SIGGRAPH'00: 517–526
8. Isenberg T, Halper N, Strothotte T (2002) Stylizing silhouettes at interactive rates: From silhouette edges to silhouette strokes. Proc. of Eurographics'02
9. Kalnins RD, Davidson PL, Markosian L, Finkelstein A (2003) Coherent stylized silhouettes. Proc. of SIGGRAPH'03: 856–861
10. Kirsanov D, Sander PV, Gortler SJ (2003) Simple silhouettes over complex surfaces. Proc. of Symposium on Geometry Processing
11. Markosian L, Kowalski MA, Trychin SJ, Bourdev LD, Goldstein D, Hughes JF (1997) Real-time nonphotorealistic rendering. Proc. of SIGGRAPH'97: 415–420
12. Northrup JD, Markosian L (2000) Artistic silhouettes: A hybrid approach. Proc. of NPAR 2000
13. Samavati FF, Bartels RH (1999) Multiresolution curve and surface representation by reversing subdivision rules. Computer Graphics Forum, Vol. 18, No. 2: 97–120
14. Sandler PV, Gu X, Gortler SJ, Hoppe H, Snyder J (2000) Silhouette clipping. Proc. of SIGGRAPH'00: 327–334
15. Sousa MC, Prusinkiewicz P (2003) A few good lines: Suggestive drawing of 3D models. Proc. of Eurographics'03: 327–340
16. Stollnitz EJ, Derose TD, Salesin DH (1996) Wavelets for computer graphics: Theory and applications. Morgan Kaufmann, San Francisco
17. van Overveld C, Wyvill B (1997) Phong normal interpolation revisted. ACM Transactions on Graphics 16(4):397–419

Cylindrical Approximation of a Neuron from Reconstructed Polyhedron*

Wenhao Lin[1], Binhai Zhu[1], Gwen Jacobs[2], and Gary Orser[2]

[1] Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA. {lin,bhz}@cs.montana.edu
[2] Center for Computational Biology, Montana State University, Bozeman, MT 59717, USA. {gwen,orser}@cns.montana.edu

Abstract. In this paper, we investigate the problem of approximating a neuron (which is a disconnected polyhedron P reconstructed from points sampled from the surface of a neuron) with minimal cylindrical segments. The problem is strongly NP-hard when we take sample points as input. We present a general algorithm which combines a method to identify critical vertices of P and useful user feedback to decompose P into desired components. For each decomposed component Q, we present an algorithm which tries to minimize the radius of the approximate enclosing cylindrical segment. Previously, this process can only be done manually by researchers in computational biology. Empirical results show that the algorithm is very efficient in practice.

1 Introduction

Approximating a neuron with cylindrical segments is an important process in constructing neural maps to study their functionality (so as to further study human/animal behavior) [5,6,8]. In practice, researchers would use commercial software to compute a polyhedron from points sampled from the surface of a neuron and then manually estimate the problem with cylindrical segments. Unfortunately, due to the complex structure of a neuron, the polyhedron reconstructed is not usually simple. In general it could be a disconnected polyhedron with extra errors (like an edge is incident to three triangles). See Figure 1 for an example of reconstructed polyhedron.

Theoretically, if we use sample points as input, i.e., given n points in 3D, compute the minimum number of cylindrical segments so as to minimize the sum of radii of these segments (or the sum of volume, or the volume of the union of the segments), then the problem is strongly NP-hard [13]. This negative result probably explains why computational biologists prefer using reconstructed polyhedron instead of sample points as input since even this erroneous polyhedron carries important geometric and topological information.

In this paper, we try to solve this problem in a semiautomatic fashion. We take a reconstructed polyhedron P, whose surface is triangulated, as input. The

* This research is supported by NSF CARGO grant DMS-0138065.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 257–266, 2004.
© Springer-Verlag Berlin Heidelberg 2004

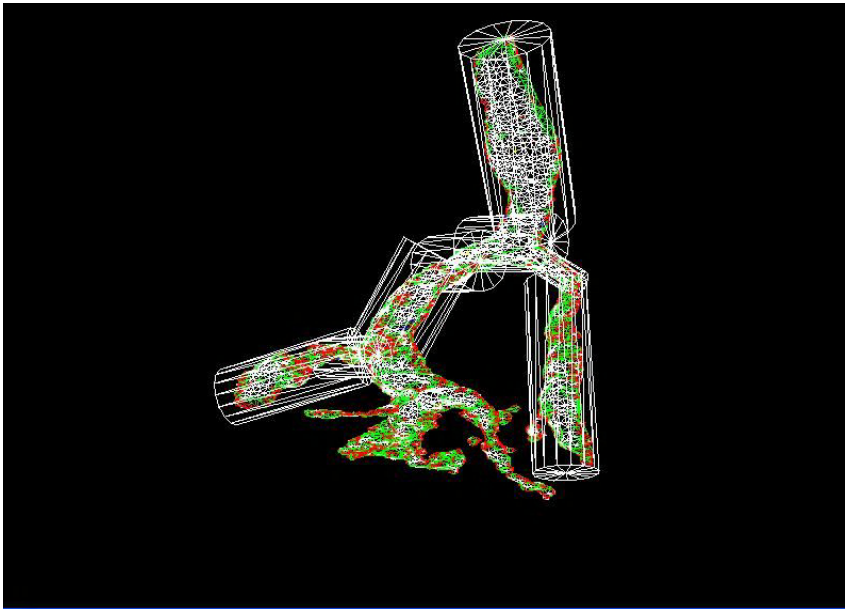


Fig. 1. A neuron polyhedron with 8954 vertices, part which is enclosed by eight cylindrical segments.

user input/feedback is minimized and is used to speed up the computation. We identify critical vertices of P around which the polyhedron encounters great geometric/topological changes (like making a turn or branching into more small branches). We then decompose P into a number of smaller polyhedra Q's. To approximate each Q, we design an efficient heuristic algorithm for computing the smallest enclosing cylinder of a set of points in 3D which does not use any complex subroutines. We also propose a method which tries to minimize the symmetric difference of volume between Q and an approximate cylindrical segment.

We first present some necessary definitions. An approximation algorithm for a minimization optimization problem Π provides a **performance guarantee** of λ if for every instance I of Π, the solution value returned by the approximation algorithm is at most λ of the optimal value for I. (*Notice that following the above definitions,* λ *is at least 1.*) For the simplicity of description, we simply say that this is a factor λ approximation algorithm for Π.

Given a set S of n points in 3D, the *diameter* of S is the maximum distance $d(p_i, p_j)$, $p_i, p_j \in S$, over all points in S. We denote it as $D(S)$. (Distances are all Euclidean unless otherwise specified.) A cylinder $\mathcal{C}$ is an infinite set of points which have at most a distance R to a given line l in 3D. The line l is called the *center* of $\mathcal{C}$. The section area of $\mathcal{C}$ which is vertical to l corresponds to a disk with radius R.

Given a segment s_1s_2 in 3D, let the distance from a point q to the line through s_1s_2 be $d(q,r)$. The distance from q to s_1s_2 is $d(q,r)$ if r is on the segment s_1s_2, otherwise the distance from q to s_1s_2 is infinite. A cylindrical segment $\mathcal{S}$ is an infinite set of points which have at most distance R to a given line segment s_1s_2 in 3D. Similarly, the segment s_1s_2 is called the *center* of $\mathcal{S}$ and R is called the *radius* of segment. The two section areas through s_1, s_2 are called the *bases* of $\mathcal{S}$. The length of s_1s_2, $d(s_1,s_2)$, is called the *length* of $\mathcal{S}$ and $2R$ is called the *width* of $\mathcal{S}$. We denote them as *length*($\mathcal{S}$) and *width*($\mathcal{S}$) respectively and we assume that *length*($\mathcal{S}$) $\geq$ *width*($\mathcal{S}$). The ratio *length*($\mathcal{S}$)/*width*($\mathcal{S}$) is called the *aspect ratio* of $\mathcal{S}$, denoted by $\alpha(\mathcal{S})$. It is easy to see that $\alpha(\mathcal{S}) \geq 1$.

2 The Smallest Enclosing Cylinder and Related Problems

In this section, we revisit the problem of computing the smallest enclosing cylinder of a set of points in 3D. This problem has found applications in CAD and CAM. Given a set of n points in 3D, Schömer et al. presented an $O(n^4\text{polylog } n)$ time algorithm [10] and Agarwal et al. obtained an $O(n^{3+\epsilon})$ time algorithm [1] to solve the problem. When n is sufficiently large, both of the algorithms are not practical. Agarwal et al. also obtained a $(1+\delta)-$approximation which runs in $O(n/\delta^2)$ time [1]. However, their algorithm used several subroutines (like computing the transversal of a set of 3D cubes and computing the smallest enclosing circle of a set of 2D points) which also make it hardly practical in terms of implementations. Recently, Chan present another $(1+\delta)-$approximation which runs in $O(n/\delta)$ time using convex programming [2]. Chan's algorithm is also not known to be practical.

In [13], an $O(n\log n + n/\delta^4)$ time algorithm is proposed to compute a $(1+\delta)$-approximation for the smallest enclosing cylinder problem. The algorithm is completely elementary and the only slightly complex subroutine is computing the diameter of the input points. The idea behind the algorithm is to first compute the diameter of the input points, which gives a rough orientation of the smallest enclosing cylinder, and then use two grids to locate the final approximate center of the smallest enclosing cylinder. However, even though it is easy to implement, for relatively large data sets with small δ it seems to be inefficient.

In this section, we first propose another $(1+\delta)$-approximation for the smallest enclosing cylinder problem. The running time is $O(n/\delta^2)$. The algorithm is easy to implement compared with previous ones. The only slightly complex subroutine used in the algorithm is to compute the smallest enclosing circle of a set of 2D points, which can be implemented efficiently [Ga].

Note that in practice the sampled points we obtain from the surface of a neuron usually contain errors. Therefore, using smallest enclosing cylinder (cylindrical segment) might not be all the biologists want. In this section, we also investigate another problem, namely, given a polyhedron Q compute a cylindrical segment C' such that the symmetric difference of volume of Q and C' is minimized. We prove a necessary condition on the optimality of C' and hence can obtain a close approximation of C'.

2.1 The Smallest Enclosing Cylinder Problem Revisited

In this subsection we first propose a simple and practical approximation algorithm for the smallest enclosing cylinder problem. Given a set S of n points, let C^* be the smallest enclosing cylinder of S (i.e., S is completely contained in C^* and the radius of C^* is minimized). Also, let the center of C^* be u^*v^*. We first have the following simple approximation algorithm:

Algorithm 1

(1) Discretize a unit ball centered at the origin into an δ by δ grid and use all unit vectors through a grid point and the origin as candidate vectors.
(2) Project all the points in S on a plane vertical to a candidate vector and compute the corresponding smallest enclosing circle of the projected points. Among all solutions, pick up the one with the smallest enclosing circle.

The idea behind this algorithm is a brute-force method: simply search over all possible discrete directions for one which has the smallest angle with u^*v^*. As computing the smallest enclosing disk of a set of n points in 2D can be done in $O(n)$ time time [3,11], it is easy to see that this algorithm runs in $O(n/\delta^2)$ time and returns a $(1+\delta)$-approximation for the smallest enclosing cylinder of S.

Notice that the radius of the enclosing cylinder of S is a function $\mathcal{F}$ of all (unit) vectors. The radius of C^* is basically the global minimum of this function. In some applications when n is large and δ has to be very small $O(n/\delta^2)$ time might not be efficient enough. Unfortunately, we do not know of any practical $O(n/\delta)$ time approximation algorithm yet. We present below a heuristic algorithm which runs in $O(n/\delta^2)$ time and always converges to a local minimum of $\mathcal{F}$. However, it seems that with practical data sampled from a neuron it runs in $O(n/\delta)$ time and almost always converges to the global minimum. We first present following simple approximation algorithm.

Algorithm 2

(1) Pick any point p' of S. Let the furthest point from p' be p".
(2) Project all points orthogonally on a plane vertical to $p'p$". Compute the smallest enclosing disk D of these projected points. Let the line passing through the center of D which is also parallel to $p'p$" be ρ. Return twice of radius of D as the width of A and ρ as the center of A. The length of A can be returned in an extra of $O(n)$ time by finding the two points furthest along $p'p$".

We have the following lemma.

Lemma 1. *Algorithm 2 presents a factor-*3 *linear time approximation for the smallest enclosing cylinder problem.*

Proof: Let the length of C^* be *length*(C^*). We consider two cases: (a) *length*(C^*)$\leq$ 2$\times$*width*(C^*), and (b) *length*(C^*)$>$ 2$\times$*width*(C^*). In case (a), it is easy to show that A has a width at most $\sqrt{5}\times$*width*(C^*). In case (b), a calculation shows that the width of A is bounded by $(2+\sqrt{1-4/\alpha^2})\cdot$*width*($C^*$),

which is always bounded by $3\times width(C^*)$. The running time of Algorithm 1 is $O(n)$ as computing the smallest enclosing disk of a set of n points in 2D can be done in $O(n)$ time [3,11]. □

In practice, especially in biology related applications, a factor-3 approximation is hardly useful. In fact, what we really want in our neuron simulation project is an approximation algorithm which runs in $O(n/\delta)$ time and returns an $(1+\delta)$-approximation. However, the only known theoretical algorithm achieving this objective [2] does not seem to be implementable. In the following we present a heuristic algorithm which always converges to a local minimum of $\mathcal{F}$. Moreover, even though its theoretical running time is $O(n/\delta^2)$, for practical data we handle in our neuron simulation project the practical running time seems to be $O(n/\delta)$. The empirical results will be presented later in this subsection.

Let C be a minimal enclosing cylindrical segment of S with center γ, i.e., when γ is fixed the radius of C is minimum, and let C_γ be the orthogonal projection of C along γ. Without loss of generality, assume that three points in S whose orthogonal projection along γ, $a_\gamma, b_\gamma, c_\gamma$, uniquely determine the smallest enclosing disk C_γ among all projected points. Let the three corresponding points in S be a, b, c.

We define the local rotation of γ toward a point a on C (but not on its bases) by ϕ as follows. We pick any two points u', v' on γ which are not contained in C such that $C \cap \gamma$ is between u', v' and $d(u', v')$ is at least $D(S)$. (Algorithmically, this can be done in $O(n)$ time by computing the minimum axis parallel bounding box $\mathcal{B}$ of S and picking up u', v' on γ such that $\mathcal{B} \cap \gamma$ is between u' and v' and $d(u', v')$ is equal to the length of the diagonal of $\mathcal{B}$. Clearly $d(u', v') = D(\mathcal{B}) \geq D(P)$.) We rotate $u'v'$ on the plane $\triangle(u'v'a)$ around u' by an angle of ϕ such that after the rotation a is closer to the new $u'v'$, i.e., the rotation is toward a (Figure 2). Symmetrically we can perform this local rotation around v'. We have the following lemma.

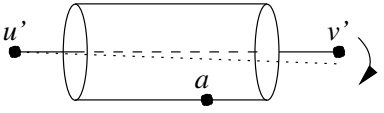


Fig. 2. Local rotation

Lemma 2. *If we can always obtain a smaller enclosing cylindrical segment by a local rotation of γ toward either a, b or c, then the radius of C is not a local optimum of $\mathcal{F}$.*

We now present the following heuristic for the smallest enclosing cylindrical segment problem.

Heuristic 1

(1) Compute the approximate cylindrical segment A of S using Algorithm 2. Let the orthogonal projection of A along its center ρ be D.
(2) Compute the minimum axis-parallel bounding box $\mathcal{B}$ of S. Let its diagonal be $D(\mathcal{B})$.

(3) Perform six possible local rotations of the center of A toward a, b and c by an angle of $\delta \cdot width(A)/3D(\mathcal{B})$. This results in 6 new directions $\rho_1, ..., \rho_6$. In each case, project S orthogonally along ρ_i and compute the smallest enclosing disk D_i. If none is smaller than D, then return D, ρ as the approximation solution. Otherwise, Update $D \leftarrow D_j, \rho \leftarrow \rho_j$ such that D_j is of the smallest size among $D, D_1, ..., D_6$ and repeat step (3).

We have the following theorem regarding Heuristic 1.

Theorem 1. *Heuristic 1 runs in $O(n/\delta^2)$ time and always converges to a local minimum of $\mathcal{F}$.*

The reason why we use Heuristic 1 is that in our neuron simulation project the cylindrical segments used are usually of large aspect ratio. Moreover, in this case, its subroutine, Algorithm 2, presents a good approximation for the center of C^*. In the following tables we compare the running times (T) and the radii (R) of the approximate smallest enclosing cylinder returned by the three $(1+\delta)$-approximations: the one proposed in [13] (which we will call Zh02 henceforth), Algorithm 1 and Heuristic 1. Table 1 shows the empirical results when the data are from our neuron simulation project. For Zh02, since the running time is virtually $O(n/\delta^4)$ so we cannot set δ to be too small as that will take several hours to run when $n \approx 2000$. So basically we compare Algorithm 1 and Heuristic 1, for which we set $\delta = 3°$. In all cases, the radius of a cylindrical segment is the measured distance (as in the table) multiplying a scaled constant (76.79/1024) millimeters. For convenience we simply use mm' as the length measure. (For example, in the fifth row, second column we have 2.543. The actual radius of the cylindrical segment is $2.543 \times 76.79/1024$ millimeters.) The data sets we use are drawn from branches in the neuron shown in Figure 1 and the platform of the testing is Java3D on which our system is built. Typically, in our application each cylindrical segment encloses less than 2000 points.

Table 1. Empirical results for Zh02, Algorithm 1 and Heuristic 1; with $\delta = 3°$ for Algorithm 1 and Heuristic 1.

n	127	233	509	1131	1479	2344	5453
$T_{Zh02}(s)$	7.735	13.906	30.609	84.484	110.563	175.781	355.578
$T_{Algorithm\ 1}(s)$	6.656	8.672	35.281	98.047	125.256	186.344	1211.609
$T_{Heuristic\ 1}(s)$	0.281	2.156	6.375	5.860	4.469	3.875	90.969
$R_{Zh02}(mm')$	2.543	6.265	6.958	9.242	9.370	12.427	27.897
$R_{Algorithm\ 1}(mm')$	2.480	6.212	6.665	8.932	9.046	12.100	27.247
$R_{Heuristic\ 1}(mm')$	2.481	6.214	6.654	9.082	8.995	12.096	27.198

We can observe that Heuristic 1 outperforms Algorithm 1 in all the 7 cases. (Notice that when $n \approx 1100$, Algorithm 1 already needs about 90 seconds to run, which is not acceptable in an interactive system.) Moreover, in 6 out of 7 cases it converges to the global minimum of the corresponding problems. (In the case when n=1131, it is not known whether Heuristic 1 converges to the

global minimum or not as we did not try a smaller δ. But even if it does not, the returned cylinder is good enough—less than 2% off the optimum). In fact, we have tested Heuristic 1 for several months over different neurons and we have yet found a practical example for which it fails (i.e., misses the global minimum by more than 5%) or takes a long time (i.e., over 10 seconds) to run.

When we change $\delta = 1^\circ$, the accuracy of Algorithm 1 does not change much. However, the running time for Algorithm 1 becomes horrible. In fact, when $n \approx 2300$ and $\delta = 1^\circ$, Algorithm 1 needs about 25 minutes to run, which makes it useless in any practical system! This makes Heuristic 1 a clear winner over Algorithm 1, even though this is not supported by the theoretical results.

2.2 The Minimal Cylindrical Segment Problem

As we mentioned at the beginning of this section, because the sample points S obtained from the surface of a neuron (hence the reconstructed polyhedron P) always contain some errors, when we decompose P into sub-polyhedra it might not be good enough to approximate a sub-polyhedron Q using the smallest enclosing cylinder of Q. A different approximation might be needed.

The minimal cylindrical segment problem is defined as follows. Given a simple polyhedron Q, compute a cylindrical segment C' such that the symmetric difference of volume between them, $Vol(Q - C') \cup Vol(C' - Q)$ is minimized. Let the part of the surface of C' which is inside Q be C^- and the part of C' which is outside Q be C^+. Also, let the area of C^- and C^+ be $A(C^-)$ and $A(C^+)$ respectively. We have the following lemma which is similar to a lemma proved in [12].

Lemma 3. *$Vol(Q - C') \cup Vol(C' - Q)$ is minimized only if $A(C^-) = A(C^+)$.*

This lemma gives us a heuristic algorithm for finding an approximate version of C', H, to approximate Q. (Notice that $A(C^-)$ and $A(C^+)$ are difficult to compute.) Assume that the direction of the center of H is given (if not, we can discretize the space to have a finite number of directions for H), then we can first compute an approximate version of H, H', such that H' is a m-prism (i.e., the section area of H' is a regular m-polygon instead of a circle) and the center of H' is the same as that of H. Certainly, this involves the computation of the intersection of H' and Q. Once H' is obtained then we simply return the smallest enclosing cylindrical segment of H' as H.

3 Decomposing a Reconstructed Polyhedron and Empirical Results

In this section, we discuss how to decompose a reconstructed polyhedron P into sub-polyhedra. We present a semi-automatic algorithm which combines the identification of critical vertices (edges) of P together with some user input.

From the user point of view, certainly an automatic solution would be the most desirable. However, due to the following reasons, for the moment this is very

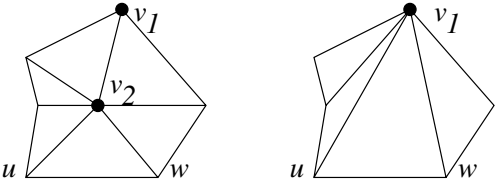


Fig. 3. An example of edge contraction: contracting (v_1, v_2) from v_2

hard to do. First, the sample points obtained from the surface of a neuron contain errors. Second, due to the complex structure of a neuron, the reconstructed polyhedron (from erroneous sample points) induce further errors. For example, in P could contain several connected components.

We now review the method in [7] to identify critical edges in a polyhedron. This method was originally proposed to simplify a polyhedron using edge contraction. Let T be a simple polyhedral surface (or polyhedron) with n vertices in 3D. Let (v_1, v_2) be an edge of T. We say that (v_1, v_2) is a *feasible* edge if contracting (v_1, v_2) from v_2 induces no self-intersection of the surface, i.e., after the edge contraction the resulting new surface is still simple (Figure 3). In addition, for each pair of triangles, $\triangle(uwv_2)$ and $\triangle(uwv_1)$, before and after the edge contraction, the angle between the outer norms of the two triangles is bounded by a constant (which is $\pi/2$ in the implementation). The latter constraint makes sure that the contraction of an edge should not change the local geometry too much. We follow [LGTW98] with the following definition.

The *weight* of a feasible edge e is defined as the product of the length of e, $|e|$, and the importance factor of e, which is related to the local geometry of e, over a reference length L_e. Intuitively, if the importance value of e is big then we should delay the contraction of e to a late stage. In [LGTW98], the importance value (or weight) of a vertex v is defined as the sharpness of v. To simplify the calculation, it is calculated as $I(v) = (x_{max} - x_{min}) + (y_{max} - y_{min}) + (z_{max} - z_{min})$ where $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$ are the minimum and maximum values of all triangle normal vectors around v. The importance factor of $e = (v_1, v_2)$ is defined as the minimum of $I(v_1)$ and $I(v_2)$. The reference length L_e is the maximum length of the axis-parallel bounding box of the model.

In [LGTW98], it was proposed that we always contract the lightest feasible edge, i.e., the one with the smallest weight as defined above. The empirical results are very promising although there is no theoretical proof that explains why an edge with a large weight is important in describing the topology of T (say, it is at the junction of two branches of T).

In our algorithm, we only use $I(v)$ defined above to classify important and common vertices in P. We set a threshold on $I(v)$'s such that all the important vertices of interest are identified. However, to achieve this, it is found that typically about 20% of the vertices are identified as well (though most of them are introduced by the errors mentioned above, i.e., most of the vertices in P with large weight should have a smaller weight if P contained less errors). Because

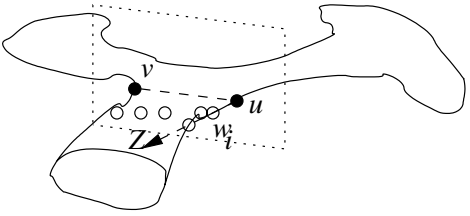


Fig. 4. Cutting a branch with user input: u, v are from user input.

of this reason, we ask the user to first input the following information to the system: (1) Connect different disconnected components to make P completely connected. (2) Around each branch where a cut should take place, input two vertices (this does require some user cooperation).

We now mention the algorithm to cut a branch our of P using the two input vertices u and v. (In our current implementation u, v are important vertices in P. But it is easy to let user choose two flat vertices and walk from those two vertices to two points with large weights.) All we want to compute is a plane which passes through u and v and cuts the branch at right angle. What we do is as follows. From u (or v) we classify all other important vertices around u into several classes. We are interested in those points $w_i, i = 1, 2, ..., W$, such that vector $\widehat{uw_i}$ and vector $\widehat{uv}$ has large angle. (Intuitively, these points are at the 'ridge' of a branch.) Let Z be the median vector of all those vectors $\widehat{uw_i}$. Then the target plane is the one passing through u, v and has the maximum angle with vector Z (Figure 4). In the system, we also give user a chance to undo this process if he/she is not satisfied with the target plane, which is usually caused by a bad choice of u and v.

Notice that in the above procedure we do not leave all the work to users by asking them to input three vertices (which uniquely defines a cutting plane). The reason is that this operation involves the rotation of P and it is always easier for user to identify two point from one side of P. Therefore, even though this procedure is simple, it is meaningful.

Our system is built on Java 3D which supports various graphics operations. The system takes a (possibly disconnected) polyhedron P and asks users to first make it connected. Then it computes all vertices with large weight. The user can input two important vertices at which he/she wants to cut a sub-polyhedron Q out of P. The system applies the above algorithm to cut Q from P. Finally the system takes Q and computes the approximate cylinder for Q using Heuristic 1 presented in Section 2. This process continues until P are completely decomposed into desirable sub-polyhedra. (Alternatively, the last loop can be implemented in a batch fashion; i.e., the user can input all pairs of important vertices first and leave all the computation to the system.)

Finally we show some empirical results. In Figure 1, we show the output of our system when we cover part of P with eight cylindrical segments. To make the image more readable, we only show a part of the approximation. (So far, for

each sub-polyhedron we compute and display the approximate smallest enclosing cylinder.) In this case, P has 8954 vertices and about twice as many faces. To manually fit this model completely, it takes a technician one to two days—it is hard to have an exact measure of time to the seconds though. With our system, it takes less than one hour.

Concluding Remarks. An interesting question is whether we could design a completely automatic method. We are currently working on smoothing a reconstructed polyhedron and on studying new methods to define critical vertices (edges).

References

1. P. Agarwal, B. Aronov and M. Sharir. Line transversals of balls and smallest enclosing cylinders in three dimensions. In *Proc. 8th ACM-SIAM Symp on Discrete Algorithms (SODA'97)*, New Orleans, LA, pages 483–492, Jan, 1997.
2. T. Chan. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. In *Proc. 16th ACM Symp on Computational Geometry (SCG'00)*, Hong Kong, pages 300–309, June, 2000.
3. J. Matoušek, M. Sharir and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1992.
4. B. Gärtner. `http://www.inf.ethz.ch/personal/gaertner/miniball.html`
5. G. Jacobs and F. Theunissen. Functional organization of a neural map in the cricket cercal sensory system. *J. of Neuroscience*, 16(2):769–784, 1996.
6. G. Jacobs and F. Theunissen. Extraction of sensory parameters from a neural map by primary sensory interneurons. *J. of Neuroscience*, 20(8):2934–2943, 2000.
7. R. Lau, M. Green, D. To and J. Wong. Real-time Continuous Multi-Resolution Method for Models of Arbitrary Topology. *Presence: Teleoperators and Virtual Environments*, 7:22–35, 1998.
8. S. Paydar, C. Doan and G. Jacobs. Neural mapping of direction and frequency in the cricket cercal sensory system. *J. of Neuroscience*, 19(5):1771–1781, 1999.
9. F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985.
10. E. Schömer, J. Sellen, M. Teichmann and C.K. Yap. Smallest enclosing cylinders. *Algorithmica*, 27:170–186, 2000.
11. E. Welzl. Smallest enclosing disks (balls and ellipsoids). In *New results and new trends in computer science*, LNCS 555, pages 359–370, 1991.
12. B. Zhu. Approximating convex polyhedra with axis-parallel boxes. *Intl. J. of Computational Geometry and Applications*, 7(3):253–267, 1997.
13. B. Zhu. Approximating 3D points with cylindrical segments. In *Proc. 8th Intl. Computing and Combinatorics Conf.(COCOON'02)*, LNCS 2387, pages 400–409, Aug, 2002.

Skeletizing 3D-Objects by Projections

David Ménegaux, Dominique Faudot, and Hamamache Kheddouci

Laboratoire LE2I – Université de Bourgogne
B.P. 47870 – 21078 Dijon cedex
{david.menegaux,dominique.faudot,hamamache.kheddouci}@
u-bourgogne.fr

Abstract. Skeletization is used to simplify an object and to give an idea of the global shape of an object. This paper concerns the continuous domain. While many methods already exist, they are mostly applied in 2D-space. We present a new method to skeletize the polygonal approximation of a 3D-object, based on projections and 2D-skeletization from binary trees.

1 Introduction

Describing an object as precisely as possible with a minimum of information is very important. The skeleton of the object (that means the principal axes of symmetry) is a solution: it gives an idea of the general aspect of the object, but ignores its thickness. In 2D-space, the skeleton is made of segments and pieces of parabolas.

On the same idea, the skeleton of 3D-space objects is made of planes and pieces of surfaces (see Figure 1). It is then difficult to obtain the exact skeleton. Methods giving an approximation of the skeleton in 3D are given for instance in [1], [14], [15].

Fig. 1. Skeleton of a 2D-polygonal shape; Skeleton of a box.

Among the several methods giving the skeleton of an object ([9], [12], [10]), we will focus on the new approach proposed in [8], based on the use of a binary or ternary tree built from the triangulation of the polygonal approximation of the object.

Our purpose in this study is to build the skeleton of a 3D-object using the projections: this will reduce the problem to 2D-space and allow us to use the rules established in [8], and then rebuild the 3D-skeleton from these results. The next section will define the basic notions and properties of skeletization, in the continuous domain. Then we will present the work produced in [8] in more in detail, in order to introduce the solution we propose in this paper to skeletize 3D-objects. We will finally give an application of it on a simple example.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 267–276, 2004.
© Springer-Verlag Berlin Heidelberg 2004

1.1 Definitions

Voronoï Diagram and Delaunay Triangulation. Let E be a finite set of points in $\Re^n$. A Voronoï region V(p, E) is associated to each point p of E, representing the nearest points of $\Re^n$ to p from any other point of E:

$$V(p, E) = \{Pi \in \Re^n / P_k \in E \setminus \{p\}, d(Pi, p) \prec d(Pi, Pk)\}. \quad (1)$$

Vor(E), the Voronoï diagram of E, is the union of the boundaries of the Voronoï regions $\delta V(p, E)$ constructed from the points p of E:

$$Vor(E) = \bigcup_{p \in E} \delta V(p, E). \quad (2)$$

The Delaunay triangulation of E is the dual of the Voronoï diagram of E [13]. This triangulation has some properties that are useful in several application areas, for instance granularity and smoothness [6].

Sampling, Sampling density, Polygonal Approximation. The sampling E_w of an object X is a set of points E extracted from the boundary of X (written δX). E is a sample of X with density w if $E \subset \delta X$ and if $\forall x \in \delta X, \exists P_i \in E$ with $d(x, P_i) < w^{-1}$. The use of sampling is emphasized by this property: raising the density w improves the distribution of the points on the boundary of X, so when w tends to infinite, the sampling tends to the boundary of the object δX.

X_w is a polygonal approximation of X with the density w if:

- the boundary of X_w is, in the plane (respectively in 3D-space), a set of simple polygons disjointed 2 by 2 (resp. a set of simple polyhedra disjointed 2 by 2 and with triangulated facets) ;
- the vertices of X_w are a sampling of δX with density w ;
- the vertices of X_w are a sampling of δX_w with density w ;
- if $d(x, \delta X) > {}^1/_w$, then $x \in X \Leftrightarrow x \in X_w$.

1.2 2D-Skeletization (Continuous Domain)

The maximum balls gives a rigorous definition of the exact skeleton of an object: a ball B included in an object X is said maximal if there is no other ball B' included in the object and strictly containing B : $B \subset B' \subset X \Rightarrow B = B'$.

Therefore, the skeleton Sk(X) of an object X in $\Re^n$ is the union of the centres of its maximum balls [11]. In the continuous domain, computing the skeleton by calculating the centres of the maximum balls is far too complicated. The other methods of skeletization have different approaches from it.

In this section, we just present two methods among those existing in the literature ([12], [10]). Let E be a set of points in $\Re^n$. At first, we must notice that:

$$Sk(\Re^n - E) = Vor(E). \quad (3)$$

These methods were chosen because they respect the criteria of convergence and reversibility [3].

Method 1 [7]: The skeleton is constructed using the Voronoï elements that are strictly included in the shape:

$$Sk_1(X_w) = \bigcup_{F \in X} \{F\}, \text{ F element of } Vor(E_w). \quad (4)$$

We finally assume the relation:

$$\lim_{w \to \infty} Sk_1(X_w) = Sk(X). \quad (5)$$

Note: with this method, the homotopy of the shape – the number of connected components – may not be preserved after skeletization (see Figure 2).

Method 2 [5]: In addition to the diagram of Voronoï, we are using the Delaunay triangulation of the polygonal approximation of the shape: the skeleton is the dual of the shape.

If $\delta X_w \subset Del(X_w)$ (i.e. the contour inclusion condition is respected), a classification of the triangles can be elaborated: the triangles strictly included in the shape, the ones strictly outside and the remaining ones called the "boundary triangles". The elements inside are a partition of the object, and we have:

$$X_w = \bigcup_{T \in X} \{T\}, \text{ T element of } Del(E_w). \quad (6)$$

The dual of the shape is made of the Voronoï elements so that their associated Delaunay elements are inside or bounded:

$$Sk_2(X_w) = Dual(X_w) \quad (7)$$

Fig. 2. (left) Method 1: the homotopy is not verified (right) Method 2: the dual of the shape

These two methods are interesting for a few reasons: the first one assures the convergence to the exact skeleton and is stuck inside of the shape, even though the homotopy may not be preserved (poor approximation of the object X); the second method grants this homotopy and the convergence, but may go outside of the shape.

2 Extracting 2D Skeleton from Trees

The purpose of our study is to find a representation of the object in 3D from a ternary tree, with this idea to keep the skeleton inside of the shape and to preserve the homotopy.

The idea of extracting the skeleton of an object from a tree was inspired by the work of [1], in which a bijection is established between a convex set of 2D points and a binary tree. This new method proposed in [8] can be divided into two steps: finding the binary tree and extracting the skeleton out of it.

2.1 Construction of the Binary Tree

Let us consider a 2D-object sampled by a set of points in order to have a polygonal approximation; then we build its Delaunay triangulation: the binary tree is, in a way, a path linking each triangle.

After choosing a face in the triangulation (in 2D-space we choose the face whose x coordinate is maximal. If two faces have the same x_{max} value, we choose the y_{max}), becoming the root of our tree, we enter the first triangle. The space is now divided into two sub-spaces – one for each face left in the triangle: they both represent a branch of the tree, letting each side be a son of the root. The node is placed in the centre of gravity of the triangle, given its properties of symmetry.

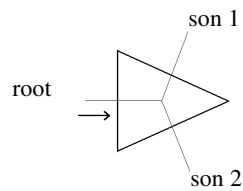


Fig. 3. Construction of a node in the triangle.

To achieve the tree, the last operation must be repeated for each triangle encountered. Each branch going outside of the convex hull is a leaf. The binary tree is not necessarily complete, and can even be degenerate.

2.2 Extraction of the Skeleton

The skeleton of the object is the binary tree without the leaves: the corresponding branches are leading outside the shape.

This process may combine the previous advantages of the two methods seen in §1.2.: being inside the shape, preserving homotopy and converging to the exact skeleton.

Note: This method works for concave shapes as well as convex shapes (see Fig. 5).

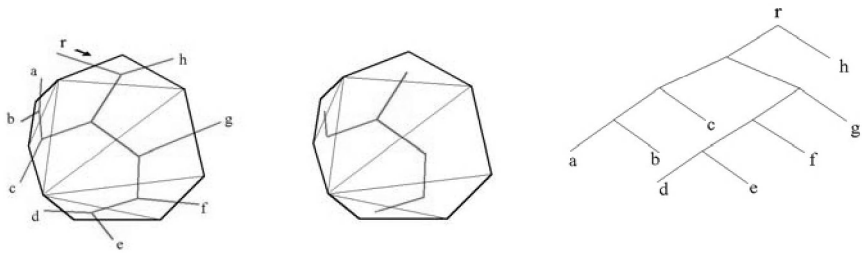


Fig. 4. Construction of the binary tree in a triangulated object: (left) we go through each triangle (middle) branches going outside are erased (right) the binary tree

Fig. 5. Complexity of the problem. (left) Initial set of points. (middle) Tetrahedrization of the set of points. (right) Ternary tree extracted from the tetrahedrization.

3 Upgrading to 3D-Objects

3.1 Tetrahedrization and Ternary Trees

The method remains the same as in 2D: at first, a tetrahedrization must be found, and then the ternary tree. We could directly extend the method to 3D objects, but we cannot extract any relevant information out of the final ternary tree because it has no topological information.

The solution we suggest uses the projections to reduce the dimension, and then uses the 2D methods. The first step is to project the object upon different planes, and get 2D-views. On each, we apply the Delaunay triangulation, extract the 2D-skeletons from which we construct a 3D skeleton.

The problem is now to find relevant projections and associate the nodes of the 2D-skeletons to have a good approximation of the 3D-skeleton.

3.2 Reducing the Dimension with Projections

We must get the polygonal approximation of the object to be skeletized. The only points to be projected are the vertices in front of the plane of projection.

The first idea in choosing the good projection would be the use of the three planes corresponding to the axes defining the 3D-space (O; x, y, z): (x, y), (x, z) and (y, z). In this operation we get a projection of the object for each dimension; but these projections are not enough to take into account every aspect of the object: we can still have hidden shape details.

As a solution, we propose to project the object on each face of the boundary cube, to obtain six views. The main advantage of this method is that we can see the main aspects of the surface of the shape. Yet, this method is not flawless, particularly on the edges and the vertices of the cube, where some imprecision may remain.

3.3 Rebuilding the 3D-Skeleton

We consider that the object has been projected on the previous boundary cube. Each of the six views contain a 2D-set of points upon which we can compute its Delaunay triangulation and the 2D-skeleton of the convex hull formed by the sets of points, using the method explianed above.

To rebuild the skeleton in 3D, we need to link the views, considering the Delaunay triangulation and locating the common sides from one view to another. We need to have a complementary description of the elements of the objects, like the Winged Edge structure [4]: it links faces, edges and vertices, and is well adjusted to our problem. Knowing the projections from the last operation and the common sides of the triangles, we can now create an assembling of triangles in 3D: when two triangles share a side in separate views, they are combined together (see Figure 6-left) and give a link between these views. Next, the vertices of the 2D-skeletons will be used. They are in the centre of gravity of each triangle. These vertices will undergo the opposite parallel projection from the last one: we get some lines of projection, one for each vertex.

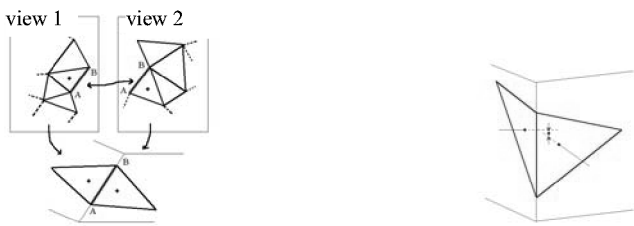


Fig. 6. (left) Assembling of triangles: a triangle in view 1 and in view 2 have a common side [AB]. These triangles are gathered (right) Construction of a vertex of the 3D-skeleton.

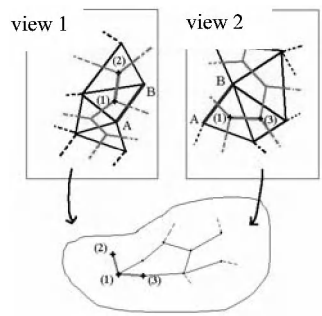


Fig. 7. Construction of the final skeleton. Between brackets, are the vertices corresponding to the 3D-version: here vertex (1) is shown up on views n°1 and n°2 because it is relative to the 3D-vertex obtained after assembling the triangles sharing edge [AB].

In many cases, the lines do not intersect in a single point. We build then one point which is equidistant from the lines, and which distance from them is minimal. This new point becomes a new vertex of the rebuild 3D-skeleton (see figure 6 right).

The operation is repeated for each assembling of triangles, to finally have a set of vertices for the 3D-skeleton. To link these vertices, we will use the 2D-skeletons again. Since the 3D-vertices are linked with the 2D-vertices, we just have to respect the 2D connexions and apply them in 3D.

The final result is a 3D-graph, approximation of the 3D-skeleton. The interests of our method are the respect of interiority criteria, the homotopy of the skeleton, the preservation of the convergence to the exact skeleton when raising the sampling density.

3.4 Application: The Cube

We consider the eight vertices of a cube and a ninth point located on one of the edges. With such a simple example, we can easily find the ternary tree made out of the tetrahedrization of the nine points, and then compare it with the result of our method.

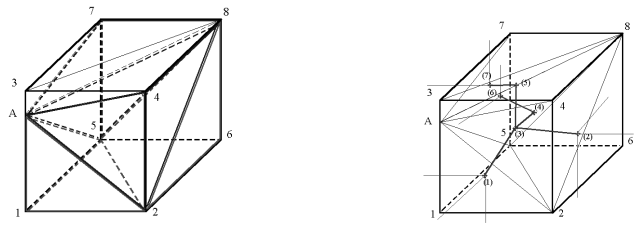


Fig. 8. Example of a cube: eight vertices with point A added on edge [13]. (left) Delaunay tetrahedrization. (right) Ternary tree of the cube (only a few leaves appear in this scheme).

Our cube now contains seven tetrahedrons:

Tetrahedrons	Vertices	Centres / nodes (see Figure 8 left)
1 internal	2A58	(3)
2 lowers	12A5, 2568	(1), (2)
2 intermediates	2A48, A578	(4), (5)
2 uppers	A348, A378	(6), (7)

The cube is projected in parallel on six planes, and then we construct the Delaunay triangulation on the result of the operation. We find:

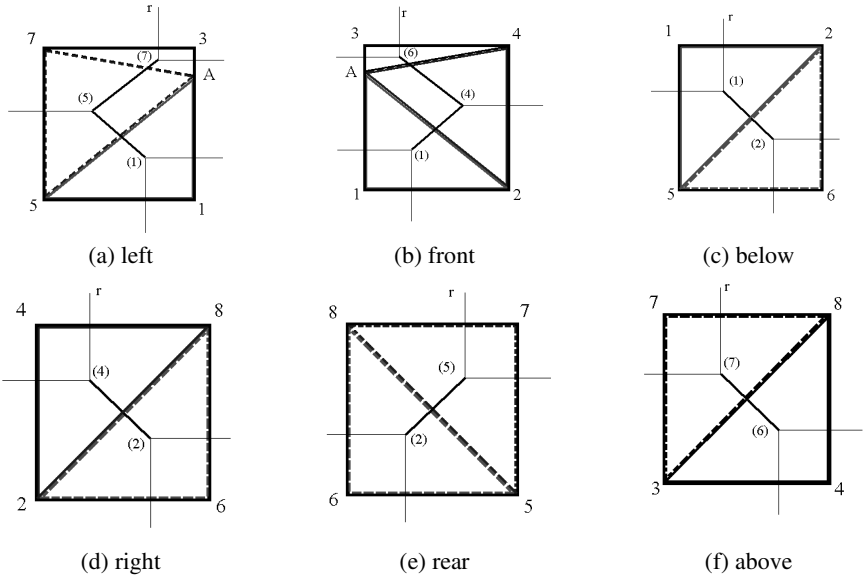


Fig. 9. Projection of the six faces of the cube. In each case, the tree gives the 2D-skeleton of the projections. Between brackets, the tetrahedrons centres seen in the last table, to which correspond the triangles.

Note: In Figure 9 (c) to (f), we only find the four vertices of a square, which are indeed co cyclic. To choose the most relevant triangulation, we base on the tetrahedrization of the initial object: for instance, tetrahedrons 12A5 and 2568 give edge [25].

At this point, we can notice that some edges are common to several views: this allows us to link some of them and to rebuild the 3D-skeleton, finding its internal nodes by associating the 2D-skeleton nodes.

At first, let us consider the triangles of the faces of tetrahedron 2568. On views (d) and (e), the triangles 268 and 568 have a common edge [68]. With (c), the same observations can be made: on the first hand 256 and 268 have [26] in common, and on the other hand 256 and 568 have [56] in common. This allows us to build an assembling of triangles (see Figure 10). With these associations, we can find one of the vertices of the 3D-skeleton. On each view, the triangles have their own centre of gravity, now projected back in parallel: the intersection gives a new vertex. In figure 14, it corresponds to the node (2) of the ternary tree built before:

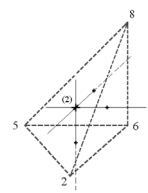


Fig. 10. Assembling triangles 256, 268 and 568, and constructing a vertex of the skeleton by projection of the centres of gravity of the triangles. This new vertex coincides with node (2) of the ternary tree.

Redoing this operation with the other triangles brings these results:
[12A, 125, 1A5] → vertex (1) ; [A57, 578] → vertex (5) ; [2A4, 248] → vertex (4)
[A34, 348] → vertex (6) ; [A37, 378] → vertex (7)

Every possible assembling has been made. We have now six vertices to construct our 3D-skeleton. To link them, we are using the 2D-skeletons. For example, on view (a), vertex (1) is linked to vertex (5), on view (b) to vertex (4) and on view (c) to vertex (2).

Following this method, we obtain a graph tending to look like the 3D-skeleton extracted from the ternary tree constructed in the beginning of this section.

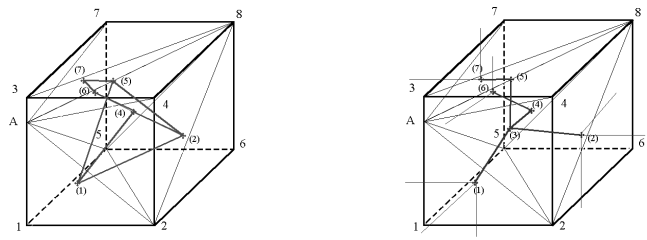


Fig. 11. (left) Final reconstitution of the 3D-skeleton. (right) the ternary tree of the cube

The skeleton built with our method is a good approximation of the exact skeleton, since it shows many similarities with the one found with the ternary tree, if we consider that the projections cannot make the tetrahedron 12A5. Therefore, the node (3) of the ternary tree has no equivalent in our result, and the connexions to it disappear.

The final result is of course an approximation of the exact 3D-skeleton of the cube, but if we refine its polygonal approximation, the result would tend to.

4 Conclusions and Prospects

In this paper we presented a new method to skeletize 3D-objects, relying on an idea recently proposed in [8] using trees. Our contribution consists in applying their method in 2D-space, to compute the skeleton of a 3D-object using projections.

This method can be decomposed in two steps. The first one is the projection of the polygonal approximation of the object on several planes in order to get different views from it. We can hence apply on each view the method of skeletization by binary trees. The second step is the reconstruction of the 3D-skeleton from the 2D-skeletons built on the last step.

Our study is limited to a restricted number of planes of projection (projection on a cube), it could be interesting to see what happens when raising this number, or even when projecting the object on a boundary sphere, to reduce the flaws. Even though, there still are some problems: the holes and concave parts of the object are not entirely (or not at all) projected with this method. Finally, the solution could be the use, in addition to the projection, of some planes of section, in order to see inside the object and exploit the new information: the trick should be then to find a method gathering properly the pieces of skeleton computed for each section.

References

1. Aldous, D.: Triangulating the Circle, at Random. American Mathematical Monthly, Vol. 101, No. 3, pp.223-233, March 1994.
2. Amenta, N., Choi, S., and Kolluri, R.: The Power Crust, SM 2001, pp. 249–260.
3. Attali, D.: Squelettes et graphes de Voronoï 2D et 3D. Phd, Grenoble1995.
4. Attali, D., Montanvert, A.: Computing and Simplifying 2D and 3D Continuous Skeletons. CVIU, Vol. 67, No. 3, pp. 261–273, 1997.
5. Baumgart, B.G.: Winged edge polyhedron representation. Technical Report CSTR-72-320, pp. 5, 1972.
6. Boissonat, J.D., Geiger, B.: Three dimensional reconstruction of complex shapes based on the Delaunay triangulation. Rapport INRIA, 1992.
7. Boissonat, J.D., Yvinec, M.: Géométrie algorithmique. Ediscience Int.l, 1995.
8. Brandt, J.W.: Convergence and continuity criteria for discrete approximations of the continuous planar skeletons. CVGIP, 59(1): 116–124, 1994.
9. Faudot, D., Rigaudière, D.: A new tool to compute 3D skeleton. ICCVG'2002, pp 258–268, 27–29 sept. 2002
10. Marion-Poty, V.: Approches parallèles pour la squelettisation 3-D. Thèse, laboratoire d'Informatique du Parallélisme, Lyon I, December 1994.
11. Ogniewicz, R., Ilg, M.: Voronoï skeletons: Theory and applications. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 63–69, 1992.
12. O'Rourke, J., Badler, N.: Decomposition of three-dimensional objects into spheres., IEEE. PAMI-1, No. 3, pp. 295–305, July 1979.
13. Schmitt, M.: Some examples of algorithms analysis in computational geometry by means of mathematic morphology techniques. LNCS, Geometry and Robotics, Vol. 391, pp. 225–246, 1989.
14. Sheehy, D.J., Armstrong, C.G., Robinson, D.J.: Computing the medial surface of a solid from a domain Delaunay triangulation. ACM Symp. on SMA, pp. 201–212, May 1995.
15. Sheehy, D.J., Armstrong, C.G., Robinson, D.J.: Shape Description By Medial Surface Construction. IEEE Trans. on Visualization and Computer Graphics, 2(1), pp. 62–72, 1996.
16. Svensson, S.: Reversible surface skeletons of 3D objects by iterative thinning of distance transforms. In G. Bertrand, A. Imiya, and R. Klette, editors, *Digital and Image Geometry*, volume 2243 of *LNCS*, pp. 395–406, 2002.

An Efficient Algorithm for Determining 3-D Bi-plane Imaging Geometry*

Jinhui Xu[1], Guang Xu[1], Zhenming Chen[1], and Kenneth R. Hoffmann[2]

[1] Department of Computer Science and Engineering
State University of New York at Buffalo
201 Bell Hall, Buffalo, NY 14260, USA.
{jinhui,guangxu,zchen4}@cse.buffalo.edu.
[2] Department of Neurosurgery
State University of New York at Buffalo
Buffalo, NY 14214, USA.
kh9@acsu.buffalo.edu

Abstract. Biplane projection imaging is one of the primary methods for imaging and visualizing the cardiovascular systems in medicine. A key problem in such a technique is to determine the imaging geometry (i.e., the rotation and translation) of two projections so that the 3-D structure can be accurately reconstructed. Based on interesting observations and efficient geometric techniques, we present in this paper a new algorithmic solution for this problem. Comparing with existing optimization-based approaches, our technique yields better accuracy, has bounded execution time, and thus is more suitable for on-line applications. Our technique can easily deal with outliers for further improving the accuracy.

1 Introduction

Effective treatment and diagnosis procedures for cardiovascular diseases heavily rely on accurate 3-D images of the interested vessel structures [8]. Because of its rapid image acquisition capability and relatively large field of view, projection imaging technique is the dominant form of imaging method, in which 3-D structures are reconstructed by using one or more 2-D projections. A key problem in such reconstructions is to determine the exact relative translation and rotation, called *imaging geometry*, of the coordinate system associated with one projection with respect to the other. Bi-Plane imaging has received considerable attention in recent years and a number of techniques have been developed for imaging geometry determination and 3-D reconstruction [2,4,7,8,9]. A common feature of these techniques is to first identify a set of correspondence points in the two projections, then convert the problem of determining imaging geometry

* The research of this work was supported in part by National Institute of Health under USPHS grant numbers HL52567. The research of the first three authors was also supported in part by an IBM faculty partnership award, and an award from NYSTAR (New York state office of science, technology, and academic research) through MDC (Microelectronics Design Center).

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 277–287, 2004.
© Springer-Verlag Berlin Heidelberg 2004

to certain non-linear optimization problem, and use either greedy approaches or general optimization packages to find a feasible solution to the imaging geometry. Due to their heuristic nature, these approaches in general guarantee neither quality of solutions nor time efficiency, and thus may not be suitable for online applications. A similar problem, called *Epipolar Geometry Determination* problem, has been studied extensively in computer vision (see the survey articles [10, 11,12]). However, almost all of them are based on iterative numerical computation which in general can not guarantee converging speed, and therefore are not suitable for online applications. Furthermore, they are all designed for the more general problem and hence cannot fully exploit the special geometric structures and properties of cardiovascular images.

To provide a better solution, we reduce the imaging geometry determination problem to the following geometric search problem: Given two sets of 2-D points $A = \{a_1, a_2, \cdots, a_n\}$ and $B = \{b_1, b_2, \cdots, b_n\}$ on two image screens (or planes) with each pair of a_i and b_i being the approximations of the two projections of an unknown 3-D point p_i, also given the 3-D coordinate system of A, find the most likely position for the origin o_B and the orientation of the coordinate system of B with respect to (the coordinate system of) A. In an ideal situation, the imaging geometry can be determined by only a constant number of correspondence pairs. In practice, however, it is often very difficult to find the exact positions for correspondence pairs (as most of the correspondences are established manually). Thus, a number of correspondence pairs are needed for ensuring the accuracy.

In this paper, we present an efficient approach for solving the above geometric search problem. Our approach first reduces the imaging geometry determination problem to a cell search problem in an arrangement of surfaces in E^6, and then simplifies the rather complicated surfaces so that each of them can be implicitly expressed by an equation. The simplified surfaces are in general non-algebraic, indicating that directly computing the arrangement could be very challenging. To overcome this difficulty, we study the error sensitivity of each variable in the imaging geometry and use it to partition the feasible domain into smaller regions so that the topological structure of the arrangement in each region can be effectively captured by some lower dimensional (e.g., 2 or 3-D) arrangements, which are relatively easy to find the optimal cells although they are still non-algebraic. Our preliminary experimental results show that the technique yields better accuracy, has bounded running time, and can be easily extended to handle outliers.

Due to the space limit, we omit lots of details from this extended abstract.

2 From Imaging Geometry to Arrangement Search

In projection imaging system, coordinate system xyz associated with beam source is related to the coordinate system uvw associated with image screen through the following formula, $u = x * D/z$, $v = y * D/z$.

To distinguish the two projections, we denote them as P_A and P_B, respectively, with P_A containing the point set A and P_B containing B. Their associated

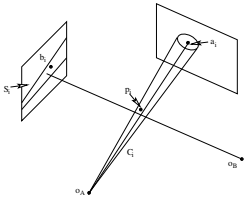


Fig. 1. Round cone.

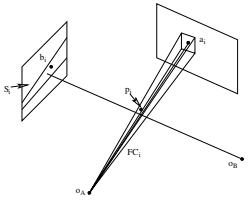


Fig. 2. Facet cone.

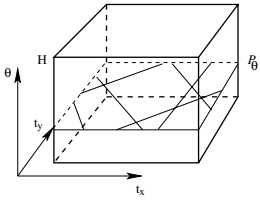


Fig. 3. Sweep the arrangement $\mathcal{A}(\Gamma) \cap H$.

image screens are denoted by S_A and S_B, respectively. We call the coordinate systems associated with P_A as xyz and uvw, and the ones of P_B as $x'y'z'$ and $u'v'w'$. The relation between the coordinate systems xyz and $x'y'z'$ can be expressed as $(x, y, z)^T = R(x', y', z')^T + \overrightarrow{t}$, where R is the rotation matrix specified by standard Euler angles, and $\boldsymbol{t}$ is the translation vector.

Due to a variety of reasons (such as movement of the beam source, data noise, and other unavoidable errors), the exact rotation matrix R and translation vector t are often unknown. Rough estimation can be obtained by using existing technique in [9]. To accurately reconstruct the 3-D structures of tiny vessels, high precision imaging geometry is desired.

Below we show how to reduce the imaging geometry determination problem to an optimal cell search problem in an arrangement of surfaces. Let $P = \{p_1, p_2 \cdots, p_n\}$ be the set of to-be-determined 3-D points. Let p_i^a and p_i^b be the exact projections of p_i on the image screen of S_A and S_B, respectively. We define $\Delta = max_{i=1}^{n} max\{dist(a_i, p_i^a), dist(b_i, p_i^b)\}$, where $dist(\cdot)$ is the Euclidean distance between two points. Note that p_i^a, p_i^b and Δ are all unknown. To determine the best possible imaging geometry G for the point set B in the coordinate system xyz, we first guess a possible value, say δ, for Δ. Clearly, if $\delta \geq \Delta$, then each p_i^a will be contained in the disk d_i (on S_A) centered at a_i and with radius δ. Thus p_i is contained in the round cone C_i apexed at the origin o_A and with d_i as the base (see Figure 1). Given a solution G to the imaging geometry of B, we can project each cone C_i to the screen of B and form a sector SC_i.

Observe that if G is optimal, then each b_i will fall in its corresponding sector SC_i. Thus, by counting the number (denoted by $f_{in}(A, B, G, \delta)$, and called *fall-in* number) of points in B which are contained in their corresponding sectors, we are able to measure the quality of G. We say G is *feasible* with respect to (w.r.t.) δ, if $f_{in}(A, B, G, \delta) = n$. For a given δ, if there exists at least one feasible G, then δ is called *feasible*. Notice that for each feasible δ, we may have infinity number of feasible solutions to G. Thus, to find the most likely imaging geometry G for B, we need not only to find a feasible solution to G, but more important to minimize the δ value, as the minimized δ value could consequently make G converge to its optimum.

Hence, to efficiently determine the imaging geometry, three problems need to be considered: (a) How to minimize δ; (b) How to determine the feasibility of δ; (c) How to find a feasible G w.r.t. a given δ.

For (a), since the feasibility of δ is monotone in the increasing direction of δ, we can perform a binary search on δ to find the smallest feasible δ, provided that we can determine the feasibility of G w.r.t. a fixed δ. For (b) and (c), we notice that given a fixed δ value, to determine the feasibility of δ and find a feasible G w.r.t. δ, it is sufficient to find a geometry G which maximizes the value $f_{in}(A, B, G, \delta)$. Hence, our focus is on this maximization problem.

Consider an arbitrary point $b_i \in B$. Let o_A^b be the projection of the origin o_A on the screen S_B, and $\boldsymbol{o_A^b b_i}$ be the ray emitting from o_A^b and crossing b_i. Let α_{b_i} be the angle between $\boldsymbol{o_A^b b_i}$ and the horizontal line (i.e., the v'-axis). Denote the lower and upper bounding rays of SC_i by r_i^l and r_i^u, respectively. Each of the two bounding rays also forms an angle with the horizontal line, and is denoted by α_i^l and α_i^u, respectively. In order for b_i to be contained in its corresponding sector SC_i (i.e., b_i contribute a "1" to $f_{in}(A, B, G, \delta)$), G must be in some positions such that α_{b_i} is between α_i^l and α_i^u. Since both r_i^l and r_i^l can be parameterized by the six variables of G, the constraint on the three angles defines a (possibly unbounded) region R_i for G in E^6 so that when G is inside R_i, sector SC_i contains b_i. Thus, in total we can generate n regions, each corresponding to a point in B. To maximize the value of $f_{in}(A, B, G, \delta)$, it is sufficient to determine a point for G in E^6 contained by the most number of R_i's.

To find a maximum point, we need to determine the bounding surface of each R_i. Notice that the bounding surface of G_i can be viewed as the loci of G while moving b_i on the two bounding rays, r_i^l and r_i^u, of SC_i. Thus, the formula of the surface can be determined by using the fact that b_i is incident to either r_i^l or r_i^u.

Once the surfaces Γ are obtained, a direct approach for computing the maximum point G is to construct the arrangement $\mathcal{A}(\Gamma)$ of Γ, and for each cell c of $\mathcal{A}(\Gamma)$ determine the value of $f_{in}(A, B, G, \delta)$. Since all points in c are contained by the same set of regions, their fall-in numbers are the same. Thus it is sufficient to consider only one point from each cell. The maximum point of G can then be determined by finding the cell with the maximum fall-in number.

3 Main Difficulties and Ideas

To make the above approach work is actually quite difficult. The success of this approach relies on two key conditions: (i) the intersections of surfaces should be easily computed, and (ii) the topological structure of the arrangement should be "simple" so that all the cells can be relatively easily constructed or detected. Unfortunately, neither one seems to be true, because to find the surface of R_i, we need to first determine the two bounding rays, r_i^l and r_i^u. They are the projections of a pair of rays on the boundary of the round cone C_i. However, r_i^l and r_i^u in general do not admit an analytical solution. Consequently, the intersection of surfaces and the arrangement cannot be efficiently computed.

To overcome the above difficulty, our idea is to approximate each round cone C_i by a convex facet cone FC_i with k facets for some small constant k (e.g., $3, 4, 6$). Depending on the location of G, the projection of FC_i will create up to k sectors, $SC_i^1, SC_i^2, \cdots, SC_i^k$, on the screen of S_B, with each sector $SC_i^j, 1 \leq$

$j \leq k$, corresponding to a pair of edges on FC_i tangent to two planes crossing o_B. The facet cone FC_i also partitions each region R_i into $O(k)$ subregions $R_i^1, R_i^2, \cdots, R_i^k$, with each subregion R_i^j generated by a sector $SC_i^j, 1 \leq j \leq k$. Since the bounding rays of each sector SC_i^j is simply the projections of a pair of pre-specified edges on the facet cone, the surface of each R_i^j can be directly determined and implicitly expressed by an equation. Let f is the angular distance between ray $\boldsymbol{o_A^b b_i}$ and one of the two bounding rays, r_i^l and r_i^u, of the sector SC_i. Let d_j denote $r_{j1}(u_{a_i} \pm \delta) + r_{j2}(v_{a_i} \pm \delta) + r_{j3}D$ for $j = 1, 2, 3$, Then we have $f = \frac{(d_2+t_y)/(d_3+t_z)-t_y/t_z}{(d_1+t_x)/(d_3+t_z)-t_x/t_z} - \frac{v'_{b_i}-t_y D/t_z}{u'_{b_i}-t_x D/t_z}$.

Using facet cones to replace the round cones although simplifies the surfaces (called *bounding surfaces*) of the regions corresponding to points in B, it introduces other problem for the arrangement. Note that each region R_i is now partitioned into $O(k)$ subregions R_i^j by a set of surfaces, called *separating surfaces*. The separating surfaces are generated by comparing the angles of the projections of the k edges of FC_i on the screen S_B, and have much more complicated form than the bounding surfaces, thus dramatically increasing the difficulty of constructing the arrangement. Notice that all bounding surfaces are still non-algebraic. Therefore, the traversing of all cells of the arrangement is very challenging. In next section, we will show that by using a different way to count the fall-in number for each cell in the arrangement, we can actually remove the set of separating surfaces. Thus we can focus on how to efficiently construct the arrangement of bounding surfaces.

To further simplify the problem, we study the error sensitivity of each variable in imaging geometry. The careful but not so difficult calculus can give us the following lemma. It shows that when the 3-D object is roughly in the middle of the image systems (which is typically the case in practice), error is much less sensitive to the three translational variables than to the rotational variables.

Lemma 1. *Let p be any point with coordinates $(x, y, z)^T$ and $(x', y', z')^T$, satisfying z and $z' \in [\frac{D}{4}, \frac{3D}{4}]$, $|x|, |x'|$, $|y|$ and $|y'| \leq \epsilon_0$ for some small constant ϵ_0. Assume that the xyz and $x'y'z'$ coordinate systems have the following relation, $\theta \leq \epsilon_1$, ψ and $\phi \in [\pi/4, 3\pi/4]$, $\phi + \psi \in [\pi/4, 3\pi/4]$, $|t_x|$, $|t_z| \in [D/4, 3D/4]$, and $|t_y| \leq \epsilon_0$, where ϵ_1 is a small constant. Then the partial derivatives of the angular distance f w.r.t. each variable has the following orders.* $\frac{\partial f}{\partial t_x} = O(\frac{1}{D^2}), \frac{\partial f}{\partial t_y} = O(\frac{1}{D})$, $\frac{\partial f}{\partial t_z} = O(\frac{1}{D^2})$, $\frac{\partial f}{\partial \theta} = O(1)$, $\frac{\partial f}{\partial \phi} = O(\frac{1}{D})$, $\frac{\partial f}{\partial \psi} = O(\frac{1}{D})$.

The above lemma shows that when p is well placed in 3-D space, the topological structure of the arrangement is more likely to change when G moves in the directions corresponding to variables with larger partial derivatives To compute the maximum point for G, we only need to find one point from each possible cell in the arrangement. Thus it is sufficient to consider a set of crossing sections (i.e., lower dimensional arrangements) of the arrangement as long as the set of crossing sections intersects every cell in the arrangement. For a non-sensitive direction (i.e., a direction with smaller partial derivative), we may select a few observing points and compute the crossing sections through the selected points.

In this way, we may avoid considering this direction continuously, and hence reduce the dimensions of the arrangement.

Hence it is possible to compute the maximum point through traversing a set of lower dimensional arrangements, if we can select 2 or 3 "good" variables with larger partial derivatives as the variables of the arrangement, and place a grid in the subspace of the domain induced by those unselected variables. We say a set of variables are good if the bounding surfaces induced by setting other variables to constants has simple forms or nice structures. Notice that in the imaging geometry determination problem, the domain can be assumed to be a small hyperbox as a rough estimation of the optimal solution can be obtained by using some previously existing techniques [9]. The sizes of the grid may vary in different directions, consistent with their partial derivatives.

4 Finding the Maximum Point in an Arrangement

To solve the maximum point problem, we need to first select the set of variables. From Lemma 1, we know that θ is the most sensitive variable to error, and thus should be chosen. Three other variables, t_y, ψ, and ϕ, has the same order. Since t_y is loosely coupled with θ, we pick t_y over the other two rotational variables.

To select other possible variables, we first observe that if two rotational variables are selected simultaneously, the surfaces will be of the form $\frac{g_1(\alpha_1,\alpha_2)}{g_2(\alpha_1,\alpha_2)}+c=0$, where, α_1 and α_2 are the two rotational variables, and $g_1(\cdot)$, $g_2(\cdot)$ are two functions containing products of trigonometric functions of α_1 and α_2. The surfaces will be rather complicated, and more importantly, their intersections will not be easily computed. Hence, in our algorithm, we only select one rotational variable. Nevertheless, a careful analysis shows that we can actually select another translational variable t_x and achieve relatively simple surfaces.

We can easily get the following two lemmas since those unselected variables are treated as constants at a fixed grid point.

Lemma 2. *Let t_y, t_x and $\alpha \in \{\theta, \psi, \phi\}$ be three selected variables. Then, at any fixed grid point, the bounding surface S_i is monotone in the directions of t_x and t_y. Furthermore, the intersection of S_i and any plane parallel to the t_xt_y plane is a straight line.*

Lemma 3. *Let t_x, t_y and α be defined as in Lemma 2. Each bounding surface S_i can be partitioned into up to three surface patches by planes parallel to t_xt_y-plane such that each surface patch is continuous in any of direction in space defined by t_x, t_y and α.*

To find the maximum point in the arrangement $\mathcal{A}(\Gamma)$, we can first use the technique in [9] to obtain an approximation of G so that the optimal solution to G is contained in an axis-aligned hyperbox H in E^6. Thus our search for the maximum point can be focused on the portion of $\mathcal{A}(\Gamma)$ inside H. In a grid point, the three unselected variables become constants, and the hyperbox H is reduced

to a 3-D axis-aligned box. Without causing any ambiguity, we also denote the 3-D box as H, the set of bounding surfaces as Γ and the arrangement as $\mathcal{A}(\Gamma)$.

Our task is to find the maximum point $\mathcal{A}(\Gamma) \cap H$. As discussed previously, all points in any cell of $\mathcal{A}(\Gamma)$ share the same fall-in number. For two neighboring cells c_1 and c_2 separated by a bounding surface S_i, the two sets of contained points in the two cells differ only by the point b_i, since crossing the surface S_i means turning b_i from contained point to non-contained point (or vice versa). Hence the difference of the fall-in numbers of the two cells is 1. To find the cell with the maximum fall-in number, our main idea is to design a plane sweep algorithm which extracts one or more points from each cell and efficiently determine their fall-in numbers.

To better illustrate our algorithm, we assume that there are only two bounding surfaces generated from each facet cone FC_i with each S_i corresponding to a bounding ray of the sector SC_i. Thus in the arrangement $\mathcal{A}(\Gamma)$, when a point G crosses S_i, its fall-in number either increases or decreases by 1. Equivalently, each surface S_i can be viewed as an oriented surface. When G crosses S_i in the direction of its orientation, the fall-in number of G increases by 1.

To efficiently search all the cells in $\mathcal{A}(\Gamma) \cap H$, we sweep a plane $\mathcal{P}$ parallel to the $t_x t_y$ plane through H. $\mathcal{P}$ starts from the bottom of H and moves in the increasing direction of θ (see Figure 3). Let $[\theta_0, \theta_1]$ be the range of θ in H, and let $\mathcal{P}_\theta$ be the intersection of $\mathcal{P}$ and $\mathcal{A}(\Gamma) \cap H$ when $\mathcal{P}$ moves to the position θ. By Lemma 2, we know that the intersections of Γ and $\mathcal{P}$ are a set of lines. Hence $\mathcal{P}_\theta$ is the portion of a straight line arrangement inside a rectangle. The following lemma shows that the fall-in number of each cell in $\mathcal{P}_{\theta_0}$ can be efficiently computed.

Lemma 4. *The fall-in number of each cell in $\mathcal{P}_{\theta_0}$ can be computed in $O(n \log n + K_0)$ time, where K_0 is the number of cells in $\mathcal{P}_{\theta_0}$.*

Proof. By using topological peeling [3], we can generate the set of cells as well as the intersections of $\mathcal{P}_{\theta_0}$ in $O(n \log n + K_0)$ time. The fall-in number of the first cell encountered by topological peeling can be computed by checking each point in B and determining whether it is contained in its corresponding sector. The time needed for checking each point is $O(1)$ once G is fixed. Thus the total time for computing the fall-in number of the first cell is $O(n)$. For each later encountered cell, we can compute its fall-in number from its neighboring cell in $O(1)$ time, since topological peeling generates cells in a wave propagation fashion. Thus the total time needed for computing fall-in numbers is $O(n + K_0)$. Thus the lemma follows.

To compute the fall-in numbers for those cells in $\mathcal{A}(\Gamma) \cap H$ which has not yet intersected with $\mathcal{P}$, we detect all the events in which $\mathcal{P}$ encounters a cell or finishes a cell while moving from bottom to top. Notice that there are several types of events which could change the topological structure of $\mathcal{P}_\theta$: (a) a surface which is previously outside H enters H and generates a line on $\mathcal{P} \cap H$; (b) a surface leaves H, and hence its corresponding line on $\mathcal{P} \cap H$ moves outside H; (c) an new cell is encountered by $\mathcal{P}$; and (d) a cell is finished by $\mathcal{P}$.

For type (a) and (b) events, we can compute for each surface $S_i \in \Gamma$ its intersections with the boundary of H, and insert the events into an event queue (such as priority queue) for the plane sweep. Since the intersections can be computed in constant time for each surface, and inserting each event into an event queue takes $O(\log n)$ time. Thus the total time for type (a) and (b) events is $O(n \log n)$ time. For type (c) and (d) events, we have the following lemma.

Lemma 5. *For any cell which is not discovered by a type (a) or (b) event, and does not intersect $\mathcal{P}_{\theta_0}$, its first intersection with $\mathcal{P}$ occurs at one of its vertices.*

Proof. By Lemma 2, we know that all surfaces in Γ are monotone in t_x and t_y directions. Suppose there is such a cell c which first intersects $\mathcal{P}$ at an interior point on one of its bounding surfaces S_i. By Lemma 3, we know S_i is continuous. Thus if move $\mathcal{P}$ up slightly, say by a sufficient small constant ϵ, then S_i will generate a closed curve on $\mathcal{P}$, thus contradicting Lemma 2.

To efficiently detect all type (c) and (d) events, let us consider a type (c) event (type (d) events can be similarly handled). Let c be the cell encountered $\mathcal{P}$. By Lemma 5, the first encountered point is a vertex v of c. Let S_1, S_2 and S_3 be the three surfaces generating v. Consider the moment just before $\mathcal{P}$ meets v. By Lemma 3, all the three surfaces S_1, S_2 and S_3 are continuous in their ranges. Thus each of them produces a line on $\mathcal{P}$. The three lines generate at least two vertices, say v_1 and v_2, on $\mathcal{P}$ which are neighboring to each other and converge to v when move $\mathcal{P}$ to v. Thus to detect this event, it is sufficient to compute v at the time when v_1 and v_2 becomes neighbor to each other at the first time.

To detect all such events, we can start from $\mathcal{P}_{\theta_0}$ and compute for each pair of neighboring vertices the moment when they converge, and store it in the event queue if it is in the range of H. Then use the event queue to sweep the arrangement. When a new vertex is generated on $\mathcal{P}$ or two vertices become neighbors to each other at the first time, we check whether there is a possible event. In this way, we can capture all the events and thus detect all the cells in $\mathcal{A}(\Gamma) \cap H$.

The following lemmas show that each type (c) or (d) event can be detected efficiently and bound the total time used for detecting all events.

Lemma 6. *The intersections of three bounding surfaces can be computed by solving a polynomial of degree* 6.

Lemma 7. *All events can be detected in $O(n \log n + T_6 K \log n)$ time, where K is the total number of vertices in $\mathcal{A}(\Gamma) \cap H$, and T_6 is the time needed for finding roots of polynomial of degree 6.*

The fall-in number of each cell c can be computed in $O(1)$ time at the moment when $\mathcal{P}$ intersects c at the first time by using the already computed fall-in numbers of its neighboring cells.

So far, we have assumed that each facet cone FC_i contributes only two surfaces to Γ. For a k-edge facet cone, it could generate k bounding surfaces,

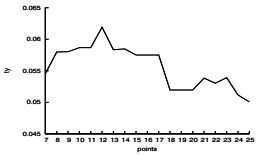


Fig. 4. Errors of t_y vs. number of corresponding pairs

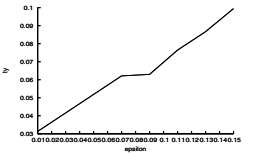


Fig. 5. Errors of t_y vs. input image errors

with each corresponding to the projection (ray) of an edge of FC_i. Let S_i^j, and r_i^j, $1 \leq j \leq k$, be the k surfaces and projection rays, respectively. Depending on the position of G, each of the k rays could be a bounding ray of the sector SC_i. When computing fall-in number, we change the fall-in number only when the surface corresponding to a bounding ray is crossed. Thus if a surfaces (called *shadow surface*) whose corresponding ray is not bounding SC_i is crossed, the fall-in number need not to be changed. As mentioned in last section, one way to solve this problem is to introduce separating surfaces, and consider the more complicated arrangement. A better way is to keep the k surfaces simultaneously in Γ, and change the way of computing the fall-in number. During sweeping the arrangement, if a new cell involves some shadow surface, then the fall-in numbers of the two cells separated by a shadow surface should be the same. A shadow surface may become a bounding surface when its corresponding ray changes roles with a bounding ray. This means the two straight lines (corresponding to the two surfaces from the same facet cone) on the sweep plane $\mathcal{P}$ intersect each other. By checking the order of the two rays on S_B, we can correctly determine which surface is now the bounding surface and its orientation.

Computing the fall-in number in this way increases the number surfaces by a factor of k. Thus the total time for finding the maximum point can be bounded by the following lemma.

Lemma 8. *The maximum point can be computed by the plane sweep algorithm in $O(nk\log(nk) + T_6K\log(nk))$ time, where k is the number of edges in a facet cone and K is the number of vertices in arrangement $\mathcal{A}(\Gamma) \cap H$ of $O(nk)$ surfaces, and T_6 is the time needed for finding roots of polynomial of degree 6.*

Lemma 9. *Let $t_a, a \in \{x, y, z\}$ and $\alpha \in \{\theta, \psi, \phi\}$ be the two selected variables, then each curve is of the form $T_a = \frac{c_i\cos(\alpha)+d_i\sin(\alpha)+e_i}{g_i\cos(\alpha)+h_i\sin(\alpha)+j_i}, \alpha \in [0, 2\pi]$ or $[0, \pi]$, where c_i, d_i, e_i, g_i, h_i and j_i are constants, and can be break into up to 3 continuous pieces. Any pair of curves have no more than 4 intersections.*

Lemma 10. *The maximum point can be found in $O(n\log n + K)$ time, where K is the number of vertices in the 2-D arrangement inside H.*

After the binary search on δ has finished, the accuracy can be further improved by removing a few outlier from the point sets A and B. Notice that

the correspondence between A and B are often done manually, and may not be consistent with each other. By removing a few outlier, we may further reduce δ and consequently reduce G. The main idea is the follows. Once δ is reduced to an infeasible value. We can find a maximum point for G, and check which point in A is not contained in its corresponding sector on S_B. If the number of such non-contained points is small, we can just throw away them from A and B. Through this way, δ is reduced. Hence, the error is reduced.

5 Experimental Results

To evaluate the performance of our technique, we implement our algorithm by using C++ and compare it with a popular approach [7] in Cardiovascular community. We conduct our experiments with the same configuration as them. Our experiments randomly generate a biplane imaging geometry at the small neighborhood of the following settings: $\psi = \pi/2, \theta = 0, \phi = 0, |t_x| = |t_z| = 0.5D, t_y = 0, D = 140cm$; and the input errors for image data are up to 0.07cm. A set of object points are placed near the center of the two systems. The object points are projected onto screen S_A and S_B, respectively. A and B are then obtained by adding some random noise to the projections of P.

Our experiments show that the absolute errors for the translation variables are as small as 0.05cm comparing to 0.15cm in [7]. The errors for the Euler angles are as small as 0.5° which is consistent with the sensitive analysis stated in Lemma 1. As expected, the following figures 4 and 5 show that the errors of t_y tend to decrease when there are more corresponding pairs, and increase when the noise of input image is higher. Similar phenomenon holds for other variables.

References

1. Amato, N.M., Goodrich, M.T., Ramos, E.A.: Computing the arrangement of curve segments: Divide-and-conquer algorithms via sampling. *Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms* (2000) 705–706.
2. S. Y. J. Chen and C. E. Metz, "Improved determination of biplane imaging geometry from two projection images and its application to three-dimensional reconstruction of coronary arterial trees," *Med. Phys.* 24: 633–654, 1997.
3. D.Z. Chen, S. Luan, and J. Xu,"Topological Peeling and Implementation," *Proc. 12th Annual International Symposium on Algorithms And Computation (ISAAC)*, Lecture Notes in Computer Science, Vol. 2223, Springer Verlag, 2001, pp.454–466.
4. J. Esthappan, H. harauchi, and K. Hoffmann, "Evaulation of imaging geometries c alculated from biplane images,"*Med. Phys.*, 25(6), 1998, pp. 965–975.
5. Z. Vlodaver, R. Frech, R. A. Van Tassel, and J. E. Edwards, Correlation of the antemortem coronary angiogram and the postmortem specimen, *Circulation 47*, pp. 162–169, 1973.
6. C. M. Grondin, I. Dyrda, A. Pasternac, L. Campeau, M. G. Bourassa, and J. Lesperance, Discrepancies between cineangiographic and postmortem findings in patients with coronary artery disease and recent myocardial revascularization, *Circulation 49*, pp. 703–708, 1974.

7. K. R. Hoffmann, C. E. Metz, and Y. Chen, Determination of 3D imaging geometry and object configurations from two biplane views: An enhancement of the Metz-Fencil technique, *Med. Phys. 22*, pp. 1219–1227, 1995.
8. K. R. Hoffmann, A. Sen, L. Lan, Kok-Gee Chua, J. Esthappan and M. Mazzucco, "A system for determination of 3D vessel tree centerlines from biplane images", *The International Journal of Cardiac Imaging 16*, pp. 315–330, 2000.
9. C. E. Metz and L. E. Fencil, Determination of three-dimensional structure in biplane radiography without prior knowledge of the relationship between the two views, *Med. Phys. 16*, pp. 45–51, 1989.
10. Z. Zhang, "Determining the Epipolar Geometry and its Uncertainty: A Review," *International Journal of Computer Vision*, 27(2): 161–195.
11. J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images-A review ," *Proc. IEEE*, Vol 76, No. 8, pp. 917–935, 1988.
12. T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review ," *Proc. IEEE*, 82(2):252–268, 1994.
13. A. Fusiello, "Uncalibrated Euclidean reconstruction: a review," *Image and Vision Computing*, Vol. 18, pp. 555–563, 2000.

Error Concealment Method Using Three-Dimensional Motion Estimation

Dong-Hwan Choi[1], Sang-Hak Lee[2], and Chan-Sik Hwang[1]

[1] School of Electrical Engineering & Computer Science, Kyungpook National University, 1370 Sankyuk-dong, Buk-gu, Daegu, 702-701, Korea
dhchoi@palgong.knu.ac.kr, cshwang@ee.knu.ac.kr
[2] School of Information & Communication Engineering, Dongyang University, 1 Kyochon-dong, Punggi-up, Youngju, Kyoungsangbukdo, 750-711, Korea
lsh@dyu.ac.kr

Abstract. A new block-based error concealment method is proposed that produces non-uniform sized and irregular quadrilateral motion estimation considering three-dimensional motions, such as rotation, magnification, and reduction as well as parallel motion, in moving pictures. The proposed error concealment method uses an affine transform, a type of spatial transform, to estimate the motion of lost block data, then the motion prediction errors are calculated using a weighting matrix and weighted according to the motion vector size for more accurate motion estimation. Experimental results show that the proposed method is able to produce a higher PSNR value and better subjective image quality by decreasing the blocking artifacts.

1 Introduction

Most video coding algorithms utilize motion compensation to exploit the temporal redundancy of the video information being sent, along with various mathematical transforms, like a discrete cosine transform (DCT), to reduce the spatial redundancy. When compressed video data are made into a bitstream and then transmitted, errors can occur in the bitstreams due to traffic congestion, channel noise, and multipath fading, etc. These bit errors mainly appear as a type of burst error and, if uncorrected, propagate in both the spatial and temporal domains, thereby seriously degrading the video quality of the received video data.

In H.263 [1] video, the basic synchronization unit is a group of blocks (GOB), so when a macroblock (MB) is corrupted, the succeeding elements in the same GOB are also discarded. As such, since uncorrectable bitstream errors in motion compensated and DCT-based video coding can cause serious degradation of the video quality, various measures against errors have already been developed [2-9]. Yet, forward error correction (FEC), which is a representative error correction method, is ineffective in a network with a limited bandwidth, as it requires a considerable amount of additive bits with a high error rate. Automatic repeat on request (ARQ) is more efficient than FEC, yet requires an additional delay

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 288–297, 2004.
© Springer-Verlag Berlin Heidelberg 2004

for the retransmission of corrupted image frames. In this case, the video decoder has to hide the visual degradation as much as possible.

Most conventional error concealment methods apply a block matching algorithm to motion estimation. However, since this algorithm presumes that a MB only moves in a horizontal or vertical direction, the resulting motion estimation is unable to produce good results with videos that include complex motions, like rotation, magnification, and reduction.

Accordingly, the current paper proposes a new block-based error concealment method that considers the three-dimensional motions of an actual image, thereby reducing the blocking artifacts present in conventional methods and enhancing the video quality of the concealed image. The proposed method uses an affine transform, a type of spatial transform, to produce a reliable approximation of three-dimensional motions based on only six parameters. Furthermore, the motion prediction error is calculated using a weighting matrix for a more accurate motion estimation of the four corners of lost MBs, plus the motion prediction error is weighted according to the magnitude of the motion vector. As such, the proposed method can be used for the robust transmission of all kinds of motion compensated and DCT-based videos headed by H.263 coded video.

2 Conventional Error Concealment Methods

In motion compensated video coding, like H.263 or MPEG, lost or erroneously received video data not only corrupts the current frame, but also propagates errors to succeeding frames. This error propagation, in both spatial and temporal directions, then results in a serious visual distortion of the output video.

Various error concealment methods have already been proposed to decrease the effect of error propagation, for example, substituting a zero motion vector for a lost one. Yet, since this method assumes less motion between consecutive frames, it is only effective for background or still images. Meanwhile, other methods use the motion vector for the same block in the previous frame or the median or average motion vectors of the available neighboring blocks [4]. However, none of these methods is appropriate for an image that includes different motions among the neighboring blocks of the lost block.

Another type of conventional error concealment method uses a block matching algorithm which is used in common motion estimation when estimating the motion of a lost block, and the movement of a MB is presumed to be in a horizontal or vertical direction, that is, a parallel motion, to make the motion estimation pertain to the same size and same quadrate shape as the lost MB. A representative example is the boundary matching algorithm (BMA) [5], which estimates a lost motion vector using the spatial correlation between the boundary pixels of the lost block and the boundary pixels of the available neighboring ones. This method first determines the variations between the current image block and the one above it, the one to its left, and the one below it, respectively, and then selects the motion vector with the smallest total variation in the three-block boundary within the search range. In this case, the chosen motion

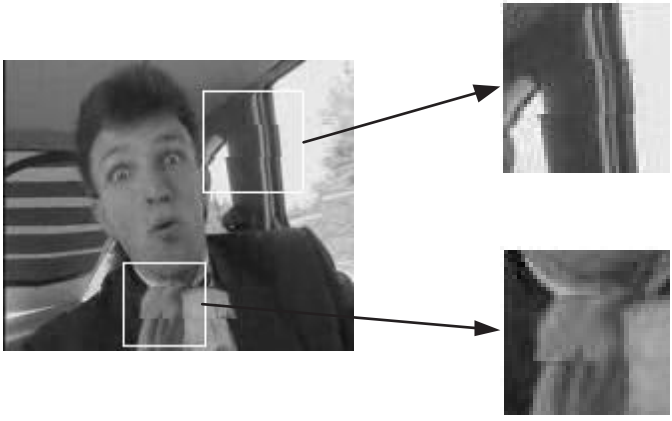


Fig. 1. Example of defects in boundary matching algorithm

vector is regarded as the optimal motion vector for the lost block. However, the image quality is still degraded when the lost block contains diagonal edges at the boundary.

In reality, most images include three-dimensional motion, where parallel motion, rotation, magnification, and reduction are all mixed owing to camera motion, like zoom-in, zoom-out, and panning, etc. or the complex motion of objects, like rotation. Consequently, a lost MB can easily have a different size and non-quadrate shape from the previous frame. Thus, if error concealment is performed through block matching motion estimation that only considers parallel motion, this can lead to serious blocking artifacts in the concealed image and degradation of the video quality owing to incorrect motion estimation. Fig. 1 shows the blocking artifacts in an image that has been error-concealed according to BMA.

3 Error Concealment Considering Three-Dimensional Image Motions

3.1 Motion Estimation Using Affine Transform

A motion model is needed that can express movement within three-dimensional space for motion estimation considering three-dimensional motions, and a predicted image is taken using a geometric transform from a previous frame. In the current paper, the geometric transform applied to change the location of a pixel in an image is an affine transform, which is a linear geometric transform used to estimate three-dimensional motions. As such, an affine transform represents a mathematical transformation of coordinates that is equivalent to a translation, rotation, expansion, or contraction with different x and y directions in relation to a fixed origin and fixed coordinate system.

Transformation equations that include complex three-dimensional motions can be expressed as

$$\begin{aligned} x' &= (x\cos\theta + y\sin\theta)S_x + T_x \\ &= (S_x\cos\theta)x + (S_x\sin\theta)y + T_x \\ y' &= (-x\sin\theta + y\cos\theta)S_y + T_y \\ &= (-S_y\sin\theta)y + (S_y\cos\theta)y + T_y \end{aligned} \tag{1}$$

where x and y are the input pixel coordinates, x' and y' are the output pixel coordinates, T_x and T_y represent shifting along the x and y axes, respectively, S_x and S_y represent scaling along the x and y axes, respectively, and θ represents the rotation angle.

By substituting the coefficients a_1, a_2, a_3, a_4, a_5, and a_6 for $S_x\cos\theta$, $S_x\sin\theta$, T_x, $-S_y\sin\theta$, $S_y\cos\theta$, and T_y, the generalized forms can be given by

$$\begin{aligned} x' &= a_1x + a_2y + a_3 \\ y' &= a_4x + a_5y + a_6 \end{aligned} \tag{2}$$

Motion estimation using an affine transform partitions an image into regional areas (blocks or patches) and estimates a set of motion parameters for each area. The process of composing a predicted image, $\hat{I}_n(x,y)$, for the nth frame from a reconstructed image, $\tilde{I}_{n-1}(x',y')$, of the (n-1)th frame can be considered as a process of texture mapping, as expressed in Eq. (3)

$$\begin{aligned} \hat{I}_n(x,y) &= \tilde{I}_{n-1}(x',y') \\ &= \tilde{I}_{n-1}(f(x,y),g(x,y)) \end{aligned} \tag{3}$$

where (x,y) and (x',y') represent the pixel coordinates corresponding to each other in the current and previous frames, respectively, and the coordinates for the previous frame can be obtained from the conversion functions $f(x,y)$ and $g(x,y)$.

Partitioning the MBs in the current frame into triangular patches and mapping them to the corresponding triangles in the previous frame produces an affine transformed motion-predicted image. In this texture mapping, the transform between two triangles is described as a two-dimensional affine transform, as expressed by the matrix equation in Eq. (4)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_4 & a_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_3 \\ a_6 \end{pmatrix} \tag{4}$$

where (x,y) and (x',y') are the pixel coordinates corresponding to each other in the current and previous frames, respectively. Obtaining the affine transform coefficients a_1 to a_6 requires the coordinates of the three triangular vertexes in the current frame and the corresponding coordinates of the three triangular vertexes in the motion-estimated previous frame, as in Eq. (5) and (6)

$$\begin{pmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix} = \begin{pmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \tag{6}$$

where x_1, x_2, x_3 and y_1, y_2, y_3 are the pixel coordinates of the triangular vertexes in the current frame, and x'_1, x'_2, x'_3 and y'_1, y'_2, y'_3 are the pixel coordinates of the triangular vertexes in the previous frame.

As such, motion estimation using an affine transform predicts the current frame using the motion vectors and texture from the previous frame, based on the following process:

- *Step 1*: The current frame image is partitioned into several triangular patches.
- *Step 2*: Motion vectors are estimated for the three vertexes based on a full search using neighboring data.
- *Step 3*: An affine transform of the triangular patches in the previous frame to their corresponding triangular patches in the current frame using the motion vectors of the vertexes produces the predicted image.

Eq. (4) and the parameters a_1 to a_6, as obtained above, can then be used to predict the locations in the previous frame that correspond to the pixels inside the triangular patches in the current frame.

In motion compensation, the intensity of the estimated locations, $\tilde{I}_{n-1}(x', y')$, can be calculated using a bilinear interpolation as follows:

$$\begin{aligned} \tilde{I}_{n-1}(x', y') = (1-\alpha)(1-\beta)\tilde{I}_{n-1}(X, Y) + (1-\alpha)\beta\tilde{I}_{n-1}(X, Y+1) \\ + \alpha(1-\beta)\tilde{I}_{n-1}(X+1, Y) + \alpha\beta\tilde{I}_{n-1}(X+1, Y+1) \end{aligned} \tag{7}$$

where (X, Y) and (α, β) are the integer and decimal part of the estimated pixel coordinates, (x', y'), respectively. The intensity values calculated from Eq. (7) are then used to reconstruct the lost blocks, thereby producing the proposed error concealment.

3.2 Error Concealment Considering Three-Dimensional Image Motions

The information of the undamaged MBs (motion vectors or reconstructed data) above and below must be used to conceal lost MBs. Here, the proposed error concealment considering three-dimensional motions of an image uses an affine transform, where the lost MBs are partitioned into two triangles and the motion vectors for their vertexes are estimated respectively.

Fig. 2 shows how undamaged data neighboring the corners of lost MBs is used to obtain their motion vectors. Among the data neighboring each corner, the neighboring undamaged data of the GOB is used for the motion estimation. Consequently, to provide a full search, the motion vectors for the corners are obtained using a search block of size C × R neighboring the corners, as shown in Fig. 2. As such, the accuracy of the data prediction for lost MBs depends on the

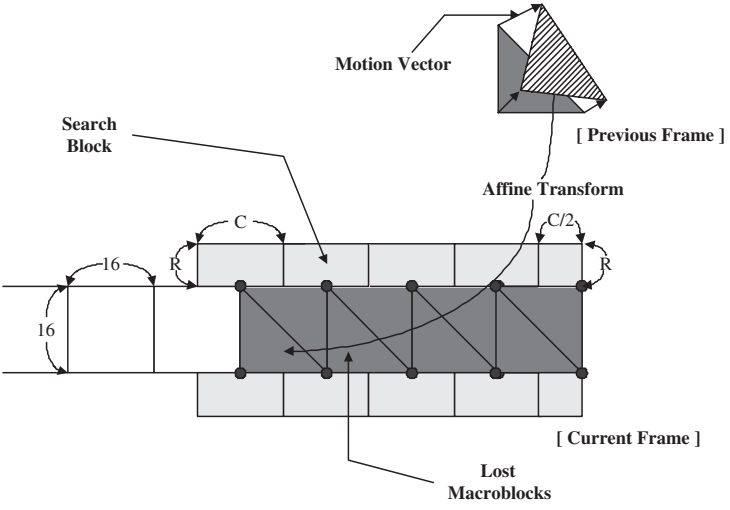


Fig. 2. Error concealment method using affine transform

accuracy of the motion vectors obtained for the corners, and the larger the search block, the more accurate the motion estimation. Therefore, the size of the search block used in the current paper was 16 × 8. In addition, the motion prediction errors are also calculated using a weighting matrix and weighted according to the magnitude of the motion vector to improve the three-dimensional motion modeling by the affine transform coefficients.

Fig. 3 shows the weighting matrix used to calculate the motion prediction error for the pixels neighboring the corners. This weighting matrix gives more weight to the motion prediction error for pixels near the corners, and less weight to those further away from the corners, where the weight values are determined based on previous experiments. That is, for more accurate motion estimation, more weight is given to the motion estimation for the corners, as this is more important than the motion estimation for the search block. The weight '4' in Fig. 3 refers to the weight of the pixels at the corners, '0' means subsampling, and the number of pixels used for motion estimation among the pixels in the search block is 128.

If the motion vectors for the corners are inaccurate, zero motion vectors will produce more efficient results. As such, if one of the motion vectors among the three triangular vertexes is wrongly estimated and its value is great, the characteristic of an affine transform means that the predicted image will be much more inaccurate than a common block matching motion estimation, thereby severely degrading the video quality of the error-concealed image. Therefore, a weight function, as in Eq. (8), is included in the motion prediction error calculation that gives preference to a lower value for the motion vector of a block corner.

$$WMSE(MV_x, MV_y) = 0.1 \times MSE(0,0) + (MV_x^2, MV_y^2) \tag{8}$$

0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
1	0	1	0	1	1	1	2	2	2	2	2	1	1	1	0	1	0	1
0	1	0	1	1	1	1	2	3	3	3	2	1	1	1	1	0	1	0
1	0	1	0	1	1	1	2	3	4	3	2	1	1	1	0	1	0	1

Fig. 3. Weighting matrix for motion estimation of block corners

When calculating the motion prediction error for a search block, if the motion vector is (MV_x, MV_y), then $WMSE$ is added to the mean square error (MSE), where $MSE(0,0)$ is the MSE obtained when (MV_x, MV_y) is $(0,0)$. In this way, after the motion vectors for the corners are obtained, the affine transform coefficients are obtained using Eq. (6), and the lost block is concealed using the predicted image obtained using Eq. (4) and (7). Here, the motion vectors for the right corners of each MB are used as the motion vectors for the left corners of the directly neighboring lost block. Thus, no blocking artifacts can occur between neighboring concealed blocks. Furthermore, since the motion vectors for the corners are obtained using data from the blocks neighboring the corners, this decreases the blocking artifacts with the neighboring blocks above and below.

4 Experimental Results

To evaluate the proposed error concealment method, the test model TMN10 [10] of ITU-T H.263+ was used. The QCIF (150 frames, Y:176 × 144, CB,CR:88 × 72) test sequences Carphone, Claire, Flower Garden, and Foreman were coded using the pattern of IPPPP$\cdots$. The PB-frame mode was not applied. As regards the coding, there were no special specifications, the QP was 5, and the frame rate was 10 frames/sec.

For the performance evaluation, the five error concealment methods shown in Table 1 were tested. As such, a comparison of the effects of error concealment was made between conventional methods, including the utilization of a zero motion vector (Zero_MV), an average motion vector of the available upper and lower blocks (Avg_MV), and a boundary matching algorithm (BMA), and the proposed method (Proposed). In addition, an error concealment method that uses the original motion vectors assuming that the motion vector information for the lost blocks is completely recovered (Org_MV) was also tested. The error concealment performance of the Org_MV is the ultimate goal for all the other error concealment methods researched.

Table 1. Error concealment methods for performance test

Method	Key algorithms
Zero_MV	copy co-sited MB from previous frame
Avg_MV	average MV of top/bottom/left/right MBs
BMA	boundary matching algorithm
Proposed	affine transform using weighting matrix
Org_MV	original MV of lost MB

Table 2. Comparison of average PSNR for different error concealment methods in objective performance test

Test sequences	Error concealment methods				
	Zero_MV	Avg_MV	BMA	Proposed	Org_MV
Carphone	28.87	29.37	27.37	31.68	32.58
Claire	34.57	34.97	31.47	37.40	37.60
Flower	17.16	21.57	18.20	23.96	23.38
Foreman	24.20	26.25	26.35	28.29	31.05

Table 3. Comparison of average PSNR for different error concealment methods in actual performance test

Test sequences		Error concealment methods					
		Loss	Zero_MV	Avg_MV	BMA	Proposed	Org_MV
Carphone	11th	12.88	29.22	31.60	29.45	33.09	32.28
	Avg	14.48	31.60	34.33	31.87	35.22	34.85
Claire	3th	13.49	34.53	34.48	31.72	37.22	37.00
	Avg	14.66	37.56	38.00	35.10	39.02	39.00
Flower	15th	11.35	21.22	25.85	22.82	28.20	27.75
	Avg	12.99	29.10	31.52	27.51	32.48	32.17
Foreman	9th	9.55	28.90	29.31	30.00	34.05	33.88
	Avg	9.91	32.43	34.73	32.90	35.94	35.98

For an objective performance test, using the 2nd to the 50th frame in each sequence, the MBs in each GOB were damaged, from the 2nd to the last MB, and from the 2nd to the 8th GOB, then error concealment was performed. Here, it was assumed that when a GOB was lost, the GOBs above and below were not lost.

Table 2 shows the average PSNR for the images that were error-concealed according to each error concealment method, based on the losslessly decoded images, rather than the original images, for the 49 frames of the four test sequences. The reason for obtaining the PSNR in this way was because the error-concealed part was used to decode the next frame, thereby allowing an evaluation of the influence of the error propagation on the next frame. The proposed error concealment exhibitd a PSNR improvement of more than 2dB compared to the other methods and a result very close to the performance of the Org_MV method.

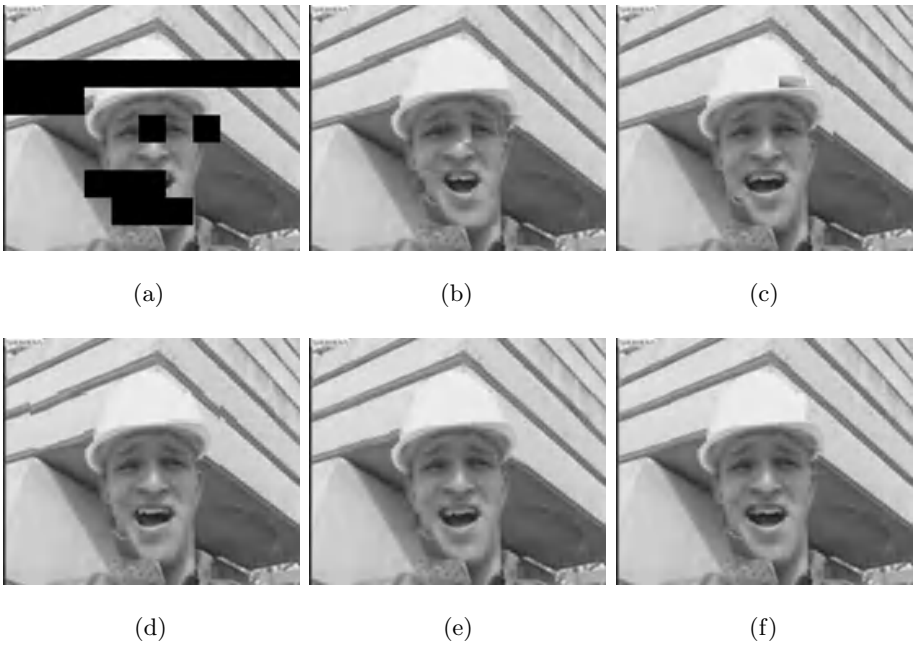


Fig. 4. Comparison of subjective image quality of Foreman sequence concealed by different error concealment methods: (a) Loss image; (b) Zero_MV image; (c) Avg_MV image; (d) BMA image; (e) Proposed image; (f) Org_MV image

To prove the performance of the proposed method against actual errors, errors were inserted in an area of the received bitstreams, then a PSNR comparison was made with the error-concealed images and the subjective image quality evaluated.

Table 3 shows the PSNR for the damaged frame and average PSNR for the frames from the damaged frame to the 50th frame during error propagation. Here, the PSNRs were obtained by comparing the error-concealed images with the original images. The Loss method means that the lost blocks were not error-concealed. The proposed method exhibited a PSNR improvement of more than 2dB in the damaged frames compared with the conventional methods, while in the frames following the damaged frame the proposed method showed a PSNR improvement of more than 1dB on average. Finally, the influence of error propagation was effectively decreased. Particularly, in the sequence Flower Garden and Carphone, the performance of the proposed method was better than that of the Org_MV method. This was because, in a sequence where the spatial redundancy of the lost area is minimal and the motion complex, block matching motion estimation is inappropriate, and the motion vectors for lost blocks used by the Org_MV method are obtained by an encoder using a block matching algorithm.

Fig. 4 shows still images of the Foreman sequence obtained using each method for a subjective evaluation of the enhanced video quality. In this case, the sub-

jective image quality of the proposed method was clearly better than that of the other conventional methods. Especially, in the areas around the mouth and hat, the proposed method produced an even better subjective image quality than the Org_MV method, not to mention the conventional methods that produced a considerable amount of blocking artifacts in the error-concealed areas.

Accordingly, the proposed error concealment method was able to produce a higher PSNR and better subjective image quality, as the motions in most video sequences generally appear as three-dimensional motions.

5 Conclusion

The current paper presented a new block-based error concealment method using three-dimensional motion estimation. The proposed method uses an affine transform, weighting matrix, and weight function for a more accurate estimation of the real motions of lost data. Experimental results confirmed that the proposed error concealment method was able to produce a higher PSNR value and better subjective image quality by decreasing the blocking artifacts. The proposed method also efficiently decreased the error propagation and even produced a better performance than the error concealment method using the original motion vectors, especially for motion regions with minimal spatial redundancy or complex motion regions.

References

1. ITU-T Recommendation H.263 Version 2: Video Coding for Low Bit-rate Communication. (1998)
2. Tsekeridou, S., Pitas, I.: MPEG-2 Error Concealment Based on Block-Matching Principles. IEEE Trans. Circuits Syst. Video Technol. **10** (2000) 646–658
3. Atzori, L., Natale, D.F., Perra, C.: Temporal Concealment of Video Transmission Errors Using Grid-Deformation Motion Model. IEE Electronics Letters **36** (2000) 1019–1021
4. Kwon, D., Driessen, P.: Error Concealment Techniques for H.263 Video Transmission. Proc. IEEE Pacific Rim Conf. on Commun., Computers and Signal Processing, (1999), 276–279
5. Lam, W.M., Reibman, A.R., Liu, B.: Recovery of Lost or Erroneously Received Motion Vectors. Proc. ICCASP, Vol. 5. (1993), 417–420
6. Al-Mualla, M., Canagarajah, N., Bull, D.R.: Temporal Error Concealment Using Motion Field Interpolation. IEE Electronics Letters **35** (1999) 215–217
7. Wang, Y., Wenger, S., Wen, J., Katsaggelos, A.K.: Error Resilient Video Coding Techniques. IEEE Signal Proc. Magazine (2000) 61–82
8. Zhang, J., Arnold, J.F., Frater, M.R.: A Cell-loss Concealment Technique for MPEG-2 Coded Video. IEEE Trans. Circuits Syst. Video Technol. **10**(2000) 659–665
9. Suh, J.W., Ho, Y.S.: Motion Vector Recovery for Error Concealment. SPIE Visual Commun. and Image Proc. (1999) 667–676
10. ITU-T Study Group 16 Version 10: Video Codec Test Model Near Terms. TMN10 (Draft 1) Document Q15-D-65 (1998)

Confidence Sets for the Aumann Mean of a Random Closed Set

Raffaello Seri and Christine Choirat

Università degli Studi dell'Insubria, 21100 Varese, Italy,
{raffaello.seri,christine.choirat}@uninsubria.it

Abstract. The objective of this paper is to develop a set of reliable methods to build confidence sets for the Aumann mean of a random closed set estimated through the Minkowski empirical mean. In order to do so, we introduce a procedure to build a confidence set based on Weil's result for the Hausdorff distance between the empirical and the Aumann means; then, we introduce another procedure based on the support function.

1 Introduction

In this paper we consider algorithms for deriving confidence regions for the mean of a sample of observed objects and shapes represented as closed and bounded (i.e. compact) sets in the Euclidean space $\mathbb{R}^d$. In order to obtain these results we rely on the powerful theory of Random Closed Sets. We suppose to observe a sample of n independent identically distributed realizations of a random element, say $\mathfrak{X}$, taking on its own values in the class of compact sets of $\mathbb{R}^d$. A precise definition of confidence region will be given in the following, but, in the meanwhile, the reader should interpret it as a region of the space containing $\mathbb{E}\mathfrak{X}$ with prescribed probability (e.g. 95%) on the basis of our sample.

Almost the same technique can be used to obtain a confidence region for a set observed with error. In both cases, the sets need not be completely observed (it is indeed enough to observe their support lines on a grid of directions); this allows, as will be discussed in the following, for applying the technique to observations derived through *computerized tomography* (see Natterer, 1986, Kak and Slaney, 1988, Gardner, 1995), *tactile sensing* and *laser-radar systems*.

In tactile sensing, a robot jaw composed of two parallel plates is clamped onto an object, thus measuring its extension in the direction perpendicular to the plates. In the 2−dimensional case, if the jaw is perpendicular with respect to the plane of the object, the lines corresponding to the plates are called "support lines". When the jaw moves with respect to the object, the support lines describe a polygonal approximation of the set. If the set is convex, this approximation can be made as precise as needed.

In laser-radar systems (or LIDAR, LIght Detection And Ranging), a laser beam is sent towards the object. The part of the beam that is reflected allows for measuring the distance between the source of the radiation and the "grazing

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 298–307, 2004.
© Springer-Verlag Berlin Heidelberg 2004

plane" (that is, the plane perpendicular to the direction of the beam and tangential to the object). In LIDAR range-and-orientation measurement, the position of the laser can vary, thus allowing for constructing a collection of planes (the grazing planes, indeed) circumscribing the shape of interest.

In computerized tomography, the support lines of the object are recorded as a subproduct of the calculation of the absorption density of the body. However, the important fact is that, in this case as well, it is possible, for a specified direction, to identify the planes passing through the extreme points of the object.

All of these applications are mostly restricted to objects in $2-$dimensional spaces: our technique though is more general since it can be used for higher dimensional Euclidean spaces (even if some limitations due to a curse of dimensionality phenomenon suggest prudent applications to problems with $d \geq 4$).

2 Some Results on Random Sets

After the first pioneering works (see Kendall, 1974, Matheron, 1975), the study of random sets has been receiving growing attention in the literature (see Goutsias, 1997, Molchanov, 1997): random sets have proved to be a valuable modelling tool in Economics, Physics and Biology and their theory offers a suitable framework to analyze old problems (e.g., epigraphical convergence in statistical estimation, see Hess, 1996, Choirat et al., 2003). The only feasible way to compare random sets (see e.g. Reyment, 1982, Stoyan and Stoyan, 1994) consists in identifying the shape of the random set with some of its measurements (length, perimeter, area, etc.) and to calculate them on a sample: clearly the choice of the measurements underlying this procedure relies heavily on the statistician's experience.

To overcome these difficulties (i.e. relying on some arbitrary shape measurements), we consider the mean of the random objects. The most convenient choice is the so-called *Aumann mean* since Central Limit Theorems for this case have already been derived. First, we need to introduce a certain number of preliminary concepts that will be used in the following.

The *distance function* from a point x to the set $C \subset \mathbb{R}^d$ is:

$$d(x, C) \triangleq \inf_{y \in C} d(x, y).$$

The *support function* of C is:

$$\mathsf{h}(y, C) = \mathsf{h}_C(y) \triangleq \sup_{x \in C} \langle y, x \rangle = \sup_{x \in C} \left(\sum_{i=1}^{d} y_i x_i \right)$$

where $y \in \mathbb{S}^{d-1} \triangleq \left\{ u \in \mathbb{R}^d : \|u\| = 1 \right\}$; it characterizes completely a closed convex set. The support function of a set $C \subset \mathbb{R}^d$ is an element of $\mathcal{C}\left(\mathbb{S}^{d-1}\right)$, the collection of continuous functions defined on the unit sphere $\mathbb{S}^{d-1}$.

The *Hausdorff distance* between two sets C and C' is defined by

$$\rho_H\left(C, C'\right) \triangleq \max \left\{ \sup_{x \in C} d(x, C'), \sup_{x' \in C'} d(x', C) \right\}.$$

The *norm* of a set C is simply:

$$\|C\| = \rho_H(C, \{0\}).$$

The *Minkowski sum* of two sets A and B is defined by:

$$A \oplus B \triangleq \{x + y : x \in A, y \in B\};$$

in the following, we will set:

$$\bar{X}_n \triangleq \frac{1}{n} \bigoplus_{i=1}^{n} X_i.$$

The *scalar multiplication* is defined as:

$$\alpha C \triangleq \{\alpha x : x \in C\}.$$

We denote by $\mathfrak{C}$ the set of all nonempty closed subsets of $\mathbb{R}^d$. Consider a set-valued map (*alias* multifunction, correspondence) $\mathfrak{X}$ from the probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to $\mathfrak{C}$. A map $\mathfrak{X}$ from Ω into $\mathfrak{C}$ is said to be $\mathcal{A}-$*measurable* if for every open subset U of $\mathbb{R}^d$, the set $\{\omega \in \Omega : \mathfrak{X}(\omega) \cap U \neq \varnothing\}$ is a member of $\mathcal{A}$. A measurable set-valued map is also called a *random closed set*, or RACS for short.

The *Aumann mean* of a RACS can be characterized through the support function as the set $\mathbb{E}\mathfrak{X}$ such that the following equality holds:[1]

$$\mathsf{h}(\cdot, \mathbb{E}\mathfrak{X}) \triangleq \mathbb{E}\mathsf{h}(\cdot, \mathfrak{X}).$$

A well known result (see Artstein and Vitale, 1975) states that RACS satisfy a Law of Large Numbers.

Theorem 1. *Let $X_1, X_2, \ldots$ be a sequence of iid random sets in $\mathbb{R}^d$ with $\mathbb{E}\|\mathfrak{X}\| < \infty$. Then*

$$\bar{X}_n \xrightarrow{\mathsf{as}} \mathbb{E}\mathfrak{X}.$$

The Aumann mean is always a convex set, but, even if the random set $\mathfrak{X}$ is not convex-valued, the Shapley-Folkman's inequality implies that the Hausdorff distance between $\frac{1}{n}\bigoplus_{i=1}^{n} X_i$ and of $\frac{1}{n}\bigoplus_{i=1}^{n} \mathsf{co}X_i$ (where $\mathsf{co}X_i$ is the convex hull of X_i) goes to 0 for large n.

A CLT for RACS can be obtained applying the CLT for $\mathcal{C}\left(\mathbb{S}^{d-1}\right)-$valued random variables (see Araujo and Giné, 1980) to the support functions.

Theorem 2. *Let $X_1, X_2, \ldots$ be a sequence of iid random sets in $\mathbb{R}^d$ with $\mathbb{E}\|\mathfrak{X}\|^2 < \infty$. Then*

$$\sqrt{n} \cdot \left(\mathsf{h}(\cdot, \mathbb{E}\mathfrak{X}) - \mathsf{h}\left(\cdot, \bar{X}_n\right)\right) \xrightarrow{\mathcal{D}} Z(\cdot),$$

where Z is a Gaussian centered process on $\mathbb{S}^{d-1}$ of covariance function $\Gamma_{\mathfrak{X}}(u, v) \triangleq \mathbb{E}Z(u)Z(v)$.

[1] This characterization holds only when the probability space is non-atomic, which is obviously the case here. See Artstein and Vitale (1975) for more details.

A fundamental result is Hörmander's formula. It relates the Hausdorff distance between sets to the $L_\infty-$distance between support functions:

$$\rho_H(C, C') = \sup_{y \in \mathsf{B}} |\mathsf{h}(y, C) - \mathsf{h}(y, C')| .$$

From this result, Weil obtains the following limit theorem for the Hausdorff distance between the empirical Minkowski mean of a sample of iid RACS and its Aumann mean.

Corollary 1. *Let $X_1, X_2, \ldots$ be a sequence of iid random sets in $\mathbb{R}^d$ with $\mathbb{E}\|\mathfrak{X}\|^2 < \infty$. Then*

$$\sqrt{n} \cdot \rho_H\left(\bar{X}_n, \mathbb{E}\mathfrak{X}\right) \xrightarrow{\mathcal{D}} \sup_{u \in \mathbb{S}^{d-1}} |Z(u)|$$

where Z is a Gaussian centered process on $\mathbb{S}^{d-1}$ of covariance function $\Gamma_{\mathfrak{X}}(u, v) = \mathbb{E}Z(u) Z(v)$.

A quick glance shows that Weil's result, stated in terms of the Hausdorff distance, is weaker than the one of Theorem 2. As already mentioned, we will develop two procedures for building confidence sets, one based on Weil's Theorem and the other on the original result on the support function. Therefore, Section 2 deals with Weil's type confidence sets and Section 3 with support function confidence sets. Section 4 presents an application to simulated data and Section 5 briefly summarizes some future developments that will be presented in a companion paper.

3 Weil's Type Confidence Sets

In the following, we suppose that the asymptotic approximation suggested by Corollary 1 is also valid for finite n; therefore, we start from equation

$$1 - \alpha = \mathbb{P}\left\{\sqrt{n} \cdot \rho_H\left(\frac{1}{n}\bigoplus_{i=1}^{n} \mathsf{co} X_i, \mathbb{E}\mathfrak{X}\right) \leq \gamma\right\};$$

remark that this is not in the form of a *confidence set*, since it is not possible to write it as:

$$1 - \alpha \leq \mathbb{P}\left\{\mathbb{E}\mathfrak{X} \subset C_\alpha\left(X_i, i = 1, ..., n\right)\right\}$$

where $C_\alpha\left(X_i, i = 1, ..., n\right)$ is the confidence set based on the sample $(X_i)_{i=1,...,n}$. However, we can write it as:[2]

[2] In the derivation, we use the characterization of the Hausdorff distance as:

$$\rho_H(C, C') = \inf\{\alpha : C \subseteq C' \oplus \alpha\mathsf{B} \text{ and } C' \subseteq C \oplus \alpha\mathsf{B}\}$$

where B is the closed unit ball.

$$
\begin{aligned}
1-\alpha &= \mathbb{P}\left\{\rho_H\left(\frac{1}{n}\bigoplus_{i=1}^{n}\mathsf{co}X_i, \mathbb{E}\mathfrak{X}\right) \le \frac{\gamma}{\sqrt{n}}\right\} \qquad (1)\\
&= \mathbb{P}\left\{\inf\left[\beta>0 : \left(\frac{1}{n}\bigoplus_{i=1}^{n}\mathsf{co}X_i\right) \subseteq \mathbb{E}\mathfrak{X}\oplus\beta\mathsf{B}\right.\right.\\
&\qquad\qquad \left.\left.\text{and } \mathbb{E}\mathfrak{X} \subseteq \left(\frac{1}{n}\bigoplus_{i=1}^{n}\mathsf{co}X_i\right)\oplus\beta\mathsf{B}\right] \le \frac{\gamma}{\sqrt{n}}\right\}\\
&\le \mathbb{P}\left\{\left(\frac{1}{n}\bigoplus_{i=1}^{n}\mathsf{co}X_i\right) \subseteq \mathbb{E}\mathfrak{X}\oplus\frac{\gamma}{\sqrt{n}}\mathsf{B} \quad\text{and}\quad \mathbb{E}\mathfrak{X} \subseteq \left(\frac{1}{n}\bigoplus_{i=1}^{n}\mathsf{co}X_i\right)\oplus\frac{\gamma}{\sqrt{n}}\mathsf{B}\right\}\\
&\le \mathbb{P}\left\{\mathbb{E}\mathfrak{X} \subseteq \left(\frac{1}{n}\bigoplus_{i=1}^{n}\mathsf{co}X_i\right)\oplus\frac{\gamma}{\sqrt{n}}\mathsf{B}\right\}. \qquad (2)
\end{aligned}
$$

Remark that this confidence set is not exact in general, that is the inequality cannot be substituted by an equality sign, not even asymptotically.

Our aim is to find an approximate value of γ from (1) and to put it in (2).

From Weil's Theorem, we have:

$$
\begin{aligned}
\sqrt{n}\cdot\rho_H\left(\frac{1}{n}\bigoplus_{i=1}^{n}\mathsf{co}X_i, \mathbb{E}\mathfrak{X}\right) &= \sqrt{n}\cdot\sup_{u\in\mathbb{S}^{d-1}}\left|\frac{1}{n}\sum_{i=1}^{n}\frac{\mathsf{h}(u,X_i)}{n} - \mathbb{E}\mathsf{h}(u,\mathfrak{X})\right|\\
&\xrightarrow[n\to\infty]{\mathcal{D}} \sup_{u\in\mathbb{S}^{d-1}}|Z(u)|,
\end{aligned}
$$

where Z is a centered random variable of $\mathcal{C}\left(\mathbb{S}^{d-1}\right)$. Therefore, for $n\to\infty$, we have:

$$
1-\alpha = \mathbb{P}\left\{\sup_{u\in\mathbb{S}^{d-1}}|Z(u)| \le \gamma\right\} = \mathbb{P}\left\{|Z|^{\max} \le \gamma\right\},
$$

where we set $|Z|^{\max} \triangleq \sup_{u\in\mathbb{S}^{d-1}}|Z(u)|$.

Since the distribution of $|Z|^{\max}$ is unknown, we introduce two approximations of this formula in order to obtain a confidence set:

1. we approximate the distribution of $|Z|^{\max}$ through the distribution of $|Z|_p^{\max}$, defined by:

$$
|Z|_p^{\max} \triangleq \max_{i=1,\dots,p}|Z(u_i)|, \quad u_i\in\mathbb{S}^{d-1},
$$

   that is the $p-$points approximation of $|Z|^{\max}$;
2. if we set

$$
\mathbf{Z} \triangleq \begin{bmatrix} Z(u_1)\\ \dots\\ Z(u_p)\end{bmatrix},
$$

we have:

$$\mathbf{Z} \sim \mathcal{N}\left[\mathbf{0}, \mathbb{V}\left(\mathbf{Z}\right)\right],$$

or equivalently

$$\zeta = \mathbb{V}\left(\mathbf{Z}\right)^{-\frac{1}{2}} \mathbf{Z} \sim \mathcal{N}\left[\mathbf{0}, \mathbf{I}\right],$$

$$\mathbf{Z} = \mathbb{V}\left(\mathbf{Z}\right)^{\frac{1}{2}} \zeta \sim \mathcal{N}\left[\mathbf{0}, \mathbb{V}\left(\mathbf{Z}\right)\right];$$

therefore, for $|Z|_p^{\max}$, (1) becomes:

$$\begin{aligned} 1-\alpha &= \mathbb{P}\left\{\max_{i=1,\ldots,p} \left|Z\left(u_i\right)\right| \leq \gamma_p\right\} \\ &= \mathbb{P}\left\{\max_{i=1,\ldots,p} |\mathbf{Z}| \leq \gamma_p\right\} \\ &= \mathbb{P}\left\{\max_{i=1,\ldots,p} \left|\mathbb{V}\left(\mathbf{Z}\right)^{\frac{1}{2}} \zeta\right| \leq \gamma_p\right\} \end{aligned}$$

unfortunately, $\mathbb{V}\left(\mathbf{Z}\right)$ is not known *a priori* and it has to be estimated through a consistent estimator, say $\widehat{\mathbb{V}\left(\mathbf{Z}\right)}$, to get

$$\widehat{\mathbb{V}\left(\mathbf{Z}\right)}^{\frac{1}{2}} \zeta = \widehat{\mathbf{Z}}$$

and

$$1-\alpha = \mathbb{P}\left\{\max_{i=1,\ldots,p} |\mathbf{Z}| \leq \gamma_p\right\} = \mathbb{P}\left\{\max_{i=1,\ldots,p} \left|\widehat{\mathbf{Z}}\right| \leq \gamma_{pn}\right\}.$$

4 Support Function Confidence Sets

Alternatively, we could obtain a confidence set for the average support function $\mathsf{h}(\cdot, \mathbb{E}\mathfrak{X})$ as the set of all the centered support functions that are upper bounded by a constant λ:

$$\begin{aligned} 1-\alpha &= \mathbb{P}\left\{\sqrt{n}\left(\mathsf{h}(u, \mathbb{E}\mathfrak{X}) - \mathsf{h}(u, \bar{X}_n)\right) \leq \lambda, \forall u \in \mathbb{S}^{d-1}\right\} \\ &= \mathbb{P}\left\{\sup_{u \in \mathbb{S}^{d-1}} \left[\sqrt{n}\left(\mathsf{h}(u, \mathbb{E}\mathfrak{X}) - \mathsf{h}(u, \bar{X}_n)\right)\right] \leq \lambda\right\}. \end{aligned} \tag{3}$$

Therefore:

$$\sup_{u \in \mathbb{S}^{d-1}} \left[\sqrt{n}\left(\mathsf{h}(u, \mathbb{E}\mathfrak{X}) - \mathsf{h}(u, \bar{X}_n)\right)\right] \xrightarrow{\mathcal{D}} \sup_{u \in \mathbb{S}^{d-1}} Z\left(u\right),$$

and we set

$$Z^{\max} \triangleq \sup_{u \in \mathbb{S}^{d-1}} Z\left(u\right).$$

We use the same approximation strategy as before:

1. since the distribution of the random variable $Z^{\max}$ is not known, we approximate it through the random variable $Z_p^{\max}$, defined as:

$$Z_p^{\max} \triangleq \max_{i=1,\dots,p} Z(u_i), \quad u_i \in \mathbb{S}^{d-1};$$

2. as before, we set

$$\widehat{\mathbb{V}(\mathbf{Z})}^{\frac{1}{2}} \zeta = \widehat{\mathbf{Z}}$$

and

$$1-\alpha = \mathbb{P}\left\{\max_{i=1,\dots,p} \mathbf{Z} \leq \lambda_p\right\} = \mathbb{P}\left\{\max_{i=1,\dots,p} \widehat{\mathbf{Z}} \leq \lambda_{pn}\right\}.$$

5 A Simulation Study

In order to present the techniques developed in the previous Sections, we analyze some simulated data.[3]

The following simulated data are generated as the convex hull of 5 points drawn from two independent standard normal variates. Figures 1 and 2 are drawn for a discretization with $p = 10$ and a sample of size $n = 50$. The dimension of the sample has the same order of magnitude than real data.

For any set X_i, $i = 1, ..., n$, a $p-$points discretized version of the support function $\mathsf{h}(x_j, X_i)$, $j = 1, ..., p$, is drawn. The mean and the variance of these functions are calculated and are used to derive, in Figure 1, the empirical cumulative distribution function of $\left|\widehat{Z}\right|_p^{\max}$ (Weil) and $\widehat{Z}_p^{\max}$ (Supp). Since their evaluation requires the integration of the density of a high-dimensional normal random vector over a rectangular domain, we have used a simulated procedure, the GHK simulator (Geweke-Hajivassiliou-Keane, see e.g. Hajivassiliou et al., 1996), with $s = 1000$ simulations in order to approximate the integral.

From these two distribution functions the values of γ_{pn} and λ_{pn} satisfying:

$$1-\alpha = \mathbb{P}\left\{\left|\widehat{Z}\right|_p^{\max} \leq \gamma_{pn}\right\},$$

$$1-\alpha = \mathbb{P}\left\{\widehat{Z}_p^{\max} \leq \lambda_{pn}\right\},$$

are obtained through an iterative procedure. In this case too, the distributions of $\left|\widehat{Z}\right|_p^{\max}$ and $\widehat{Z}_p^{\max}$ have been approximated through the GHK simulator, with

[3] The following simulations have been programmed in R, a free software that can be downloaded from `http://www.r-project.org`. See Ihaka and Gentleman (1996).

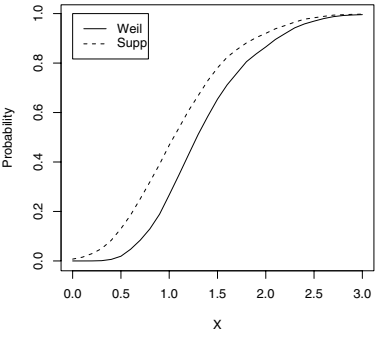


Fig. 1. The cumulative distribution function of $\left|\widehat{Z}\right|_p^{\max}$ (Weil) and $\widehat{Z}_p^{\max}$ (Supp)

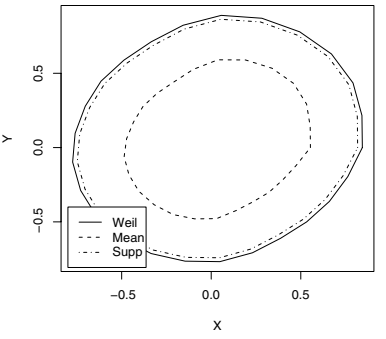


Fig. 2. Minkowski mean (Mean) of a sample and confidence sets (Weil, Supp) for the Aumann mean

$s = 30$ simulations. These values are then used in order to obtain confidence sets as described in Sections 2 and 3. In Figure 2, Mean is the Minkowski mean of the sample, Weil is the confidence set described in Section 2 and Supp is the confidence set of Section 3. It is evident, even from this example, that the procedure based on the support function yields smaller confidence set than the one based on Weil's Theorem 1.

Clearly, a simple inspection of the formulas should convince the reader that the values of γ_{pn} and λ_{pn} are expected to increase with p. Table 1 shows the values of γ_{pn} and λ_{pn} for different p and n, where the number of simulation s is fixed to 30 and the number of replications to 200.

Figure 3 shows the kernel estimators based on a bandwidth of 0.15 of the densities of γ_{pn} and λ_{pn} for $p = 12$ and $n = 50$, 30 simulations and 200 replica-

Table 1. γ_{pn} and λ_{pn} for confidence sets (standard errors in parentheses)

(p,n)	25	50	100
γ_{pn}			
10	2.244608 (0.2498404)	2.253287 (0.2095767)	2.230253 (0.1877489)
30			2.23936 (0.2307905)
λ_{pn}			
10	2.041577 (0.2600087)	2.042199 (0.2019208)	2.031207 (0.1811784)
30			2.045988 (0.2263206)

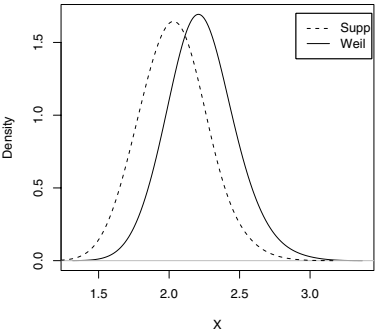


Fig. 3. Kernel estimators of the density of λ_{pn} (Supp) and γ_{pn} (Weil)

tions. It is evident from the data that the confidence set based on the support function is smaller than the one based on the Hausdorff distance.

6 Further Developments

A companion paper (see Choirat and Seri, 2003) shows that the previous procedures, for $p, n \to \infty$, yield consistent confidence sets for the Aumann mean of a RACS and establishes their rates of convergence: moreover, the confidence set based on the support function is shown to dominate strictly the one based on the Hausdorff distance. At last, a limited simulation study illustrates the feasibility and the precision of the present approaches.

References

Araujo A., Giné E.: The central limit theorem for real and Banach valued random variables, Wiley, New York (1980)

Artstein Z., Vitale R.A.: A strong law of large numbers for random compact sets. The Annals of Probability **3** (1975) 879–882

Choirat C., Seri R.: Confidence sets for the Aumann mean of random closed sets. Working Paper, Université Paris 9 Dauphine (2003)

Choirat C. ,Hess C., Seri R.: A Functional Version of the Birkhoff Ergodic Theorem for a Normal Integrand: A Variational Approach. The Annals of Probability **31** (2003) 63–92

Gardner R.J.: Geometric tomography, Encyclopedia of mathematics and its applications **58** Cambridge University Press (1995)

Goutsias J.: Morphological analysis of random sets, an introduction. In: Random sets, theory and applications, J. Goutsias, R.P. Mahlher, H.T. Nguyen. eds, Springer (1997) 2–26

Hajivassiliou V., McFadden D.L., Ruud P.: Simulation of multivariate normal rectangle probabilities and their derivatives: Theoretical and computational results. Journal of Econometrics **72** (1996) 85–134

Hess C.: Epi-convergence of sequences of normal integrands and strong consistency of the maximum likelihood estimator. The Annals of Statistics **24** (1996) 1298–1315

Ihaka R., Gentleman R.: R: a language for data analysis and graphics. Journal of Computational and Graphical Statistics **5** (1996) 299–314

Kak A.C., Slaney M.: Principles of computerized tomographic imaging, IEEE Press (1988)

Kendall .G.: Foundations of a theory of random sets. In: Advances in theory and applications of random sets, E.F. Harding, D.G. Kendall eds, Wiley, London (1974) 322–376

Matheron G.: Random sets and integral geometry, Wiley, New York (1975)

Molchanov I.S.: Statistical models for random sets. In: Random sets, theory and applications, J. Goutsias, R.P. Mahlher, H.T. Nguyen eds, Springer (1997) 27–45

Natterer F.: The mathematics of computerized tomography, Wiley, Stuttgart (1986)

Reyment R.A.: Multivariate Morphometrics. In: Handbook of Statistics, Volume 2, P.R. Krishnaiah and L.N. Kanal eds, North-Holland Publishing Company (1982) 721–745

Stoyan D., Stoyan H.: Fractals, random shapes and point fields, Wiley, Chichester (1994)

Weil W.: An application of the central limit theorem for Banach-space-valued random variables to the theory of random sets. Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete **60** (1982) 203–208

An Algorithm of Mapping Additional Scalar Value in 2D Vector Field Visualization

Zhigeng Pan[1,2], Jianfeng Lu[2], and Minming Zhang[2]

[1] Institute of VR and Multimedia, HZIEE, Hangzhou, 310037, China
[2] State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310027, China
(zgpan,jflu,zmm)@cad.zju.edu.cn

Abstract. Vector field visualization is the most challenging task in the scientific visualization. The algorithm of Line Integral Convolution based on texture image can depict the details of the vector field. To Display multi-dimensional information on the output image, Some methods such as color mapping, tone mapping are developed to show the direction, orientation and magnitude of the vector field. In this paper, we propose a new method that mapping additional scalar values to the local contrast of the output texture.

Keywords: Scientific Visualization, Multivariate visualization, Line Integral Convolution, Image Contrast

1 Introduction

Vector field visualization is the most important research task of the scientific visualization. A graphical representation of the data can make the observer infer the types and the distributions of objects from a given pattern. The traditional visualization approach, such as graphic icons, streamlines, particle traces, needs to choose the seed points carefully to avoid losing details of the field. To solve this problem the image based methods are developed such as spot noise [1] and LIC [2]. Based on this technique, many research works involve in improving the texture quality and decreasing the calculation time [3, 10]. Moreover, the dataset in scientific computation contains multidimensional information; for instance, there are several scalar values such as temperature, pressure etc in one point of flow field. It is a challenging work for us to map multidimensional values into output texture. Colors are often used to map one dimension value but it is not sufficient. Bump mapping technique is used to map additional values in Sanna [9]. In addition, Sanna [10] notices the sensitivity of the human eye to different contrast levels and maps scalar values to the image contrast. The method in his paper is based on adjusting the parameters in LIC algorithm. We propose a new method mapping scalar values to local image contrast based on texture mapping and image process. The approach is robust and easy to implement.

The paper is organized as follows: we first discuss previous relevant work in section 2 and in section 3 we describe our algorithm in the details. Some examples are shown in section 4. Finally, remarks and conclusion can be found in section 5.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 308–314, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Previous Work

In the history of vector field visualization, texture based method was firstly proposed by Van Wijk [1] which is called Spot Noise. The method filters the sample points along the vector field direction and creates the finally image with strong correlation in the direction. Brian Cabral and Leith Leedom [2] introduced the Line Integral Convolution(LIC) algorithm which convolve a white noise texture along a path of vectors tangent to the field. The output texture of LIC is pixel resolution and depicts the vector field direction clearly. We also can change the phase of the convolution kernel to create animation to display the orientation.

Given a vector field defined by $v:\Re^2 \to \Re^2, x \mapsto v(x)$, To point x_0, the output intensity is calculated as following: First we integrate a streamline $\sigma(u)$ throughout x_0, then doing a re-parameterization of $\sigma(u)$ by arc-length s, after sample the points in white noise image, we can calculate the intensity $I(x_0)$ followed the equation (1):

$$I(x_0) = \frac{1}{\int_{-L}^{L} K(s)ds} \int_{s_0-L}^{s_0+L} K(s-s_0)T(\sigma(s))ds \tag{1}$$

$T(x)$ is an white noise texture, $K(s)$ is the kernel function, L is the half of the integral length.

Based on the original LIC algorithm, many improved approach are proposed Detlev Stalling [3] developed FastLIC method to improve the speed ten times, by reuse the calculation result of the neighbor pixels. Lisa Forssell [5] extended the algorithm to the arbitrary curvilinear grid surface by transferring from compute space to physical space. An extended LIC called UFLIC proposed by H.W. Shen [6], is successfully applied in visualizing unsteady flow fields. In the aspects of mapping additional scalar values to texture image, using colors to denote vector field property is a classic method. H.W. Shen [7,8] uses dye technique to enhance flow feature in his UFLIC algorithm. Sanna [9] combines LIC and bump mapping technique to bump and depress the tone of output texture according to the scalar value where to be mapped. In his another paper (Sanna [10]), he uses local contrast level to denote information in a texture by adjusting the algorithm parameters according to the additional scalar values.

3 The Mapping Algorithm

Our goal is to use different level of local contrast to denote the additional scalar values. We use a completely different approach from Sanna [10]. Considering in a small region in the stable vector field, the streamlines are approximately parallel to each other. We preprocess a series of LIC like textures with different contrast and then map them to the region according to the scalar values. After that, we adjust local contrast adaptively to enhance the texture image as the image post process to get the better result. We describe the algorithm in details as following:

3.1 Small Region Texture Mapping

In our algorithm, we use box convolution kernel $K(s)$ to calculate LIC algorithm. We present a discretized version:

$$I(x_0) = \frac{1}{2n+1}\left(\sum_{i=-n}^{n} T(x_i)\right) \tag{2}$$

As Illustrated in figure 1 we forward and backward sample n points along the integral path σ from the start point x_0 which to be calculated. $T(x_i)$ is the sample value of the input texture along the path. According to Fast LIC [3], when we calculate point x_1, there are $2n-1$ points are the same from x_1's sample points to x_0's. So we modify equation 2 to the difference formula as equation (3):

$$I(x_{i\pm1}) = I(x_i) + \frac{1}{2n+1}[T(x_{i\pm(n+1)}) - T(x_{i\mp n})] \tag{3}$$

Because the input texture is a white noise image, the part

$$\frac{1}{2n+1}[T(x_{i\pm(n+1)}) - T(x_{i\mp n})]$$

can be considered as a small random value. This simplification will not affect the output image much but can speed up our calculation. Using this method, we pre-compute the output pixels of the whole streamline and map them to the output texture without calculating them again. Considering streamlines in the small region can be regarded as parallel approximately, we extend line to a region. In figure 1, when we calculate point x_0, we just map pre-computed texture to the small region abcd.

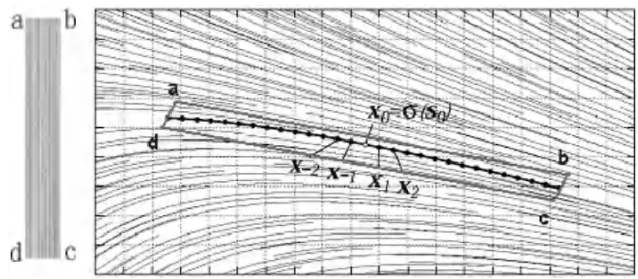


Fig. 1. Sample and texture mapping

A C-like pseudo code of our basic algorithm is shown as follow:

```
For Pixels In Output Image
    If Pixel[i] Not Caculated Then
        MapTextureToImage();
        /* we accumulate intensity of the pixels which are hit more times */
    End If
NormalizeOutputImage()
/* we use eqution  V=Accu_Value/Hit_Times to calculate the pixels  which are hit
more times. Accu_Value: the sum of the intensity value. Hit_Times: Hit times of the
pixels.*/
```

As pointed out in FastLIC [3], we use SOBOL sequence to loop the pixels in the output texture image. This can improve the numbers of the pixels to be hit and decrease the total calculation time.

3.2 Create Texture with Different Level of Local Contrast

It has been proved that the level of contrast strongly affects the capability of human eye/brain system to perceive details [10]. So when we create pre-computed texture image, we dynamical adjust the local contrast of each image according to the additional scalar values of the vector field. That is, areas denoted by larger scalar values will have higher contrast levels and the zones where contains small scalar values will be represented by lowly contrast image. We define the local contrast as equation (4):

$$C = \frac{|\Delta L|}{L} \tag{4}$$

ΔL: The difference intensity between the pixel and the background. L: The intensity of the background.

We use equation (5) to calculate the intensity of the streamlines in texture image.

$$C_{line} = C_L + [Rand(K_{Scale}) - K_{Scale}/2] \tag{5}$$

C_L: Intensity of the background, K_{Scale} is the difference between the pixels intensity with the background intensity. *Rand*() is the random function.

When K_{Scale} varies from [10~80], the computed texture images are shown as following figure (2):

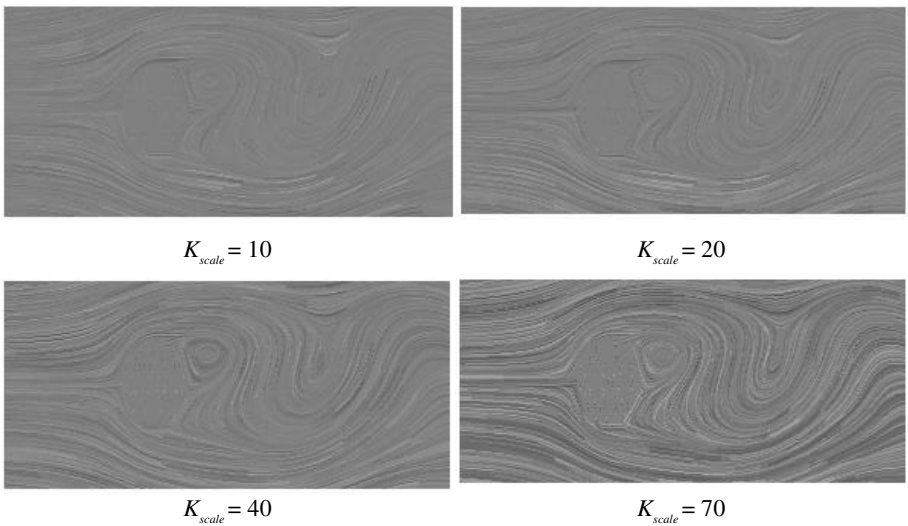


Fig. 2. Pre-computed texture with different local contrast

Passing through the two steps described above, to prevent pixels from blurring caused by multi times hit we adopt adaptive local contrast adjusting as the image post process. The basic idea is to adjust the intensity of a pixel according to the scalar values with adaptive coefficient.

$$ACE = K_1\left[\frac{m_{I(r,c)}}{\sigma_I(r,c)}\right]\left[I(r,c) - m_I(r,c)\right] + K_2 m_I(r,c) \tag{6}$$

$m_{I(r,c)}$: The average intensity of the texture image, $\sigma_{I(r,c)}$: The local intensity variance of the pixels in slide window centered in current pixel, $m_I(r,c)$: The local average intensity of the pixels in slide window centered in current pixel K_1,K_2: Scale coefficients

The part $\left[\frac{m_{I(r,c)}}{\sigma_I(r,c)}\right]$ of equation (6) is the adaptive coefficient which is lower in the high contrast place and higher in low contrast place.

The flow chart of our algorithm is shown as figure (3):

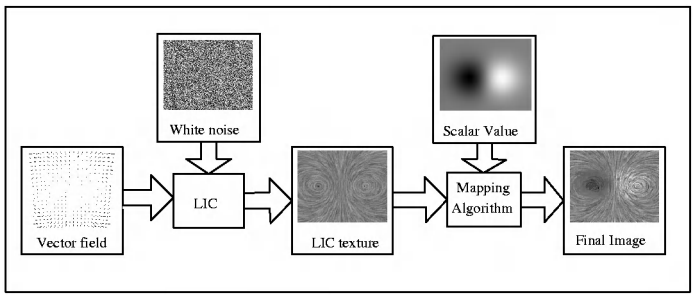


Fig. 3. Algorithm schema

4 Examples

We give some examples computed by our algorithm in figure (4). The flat form is: PIII667 Mhz CPU, 256M Memory, Gf4 64M Display Card. The image resolution is 512X512 and calculation time is about 5-6s.The vector field is created by numerical method. The first example (figure A_1, A_2) shows vector field of double vortex. The second and last examples (figure A_3-A_6) show a vector field represent by polynomial expression. The left side image is additional scalar values. The scalar values in figure B_1, C_1 refer from Sanna [9,10]. Our algorithm is to use the local contrast level to denote the scalar values (figure left side) in output texture (figure right side). It is clearly that in the areas with large scalar values white part in the left side scalar values, the image contrast of corresponding part on the right side is enhanced. On the other hand, areas with lower values is depressed.

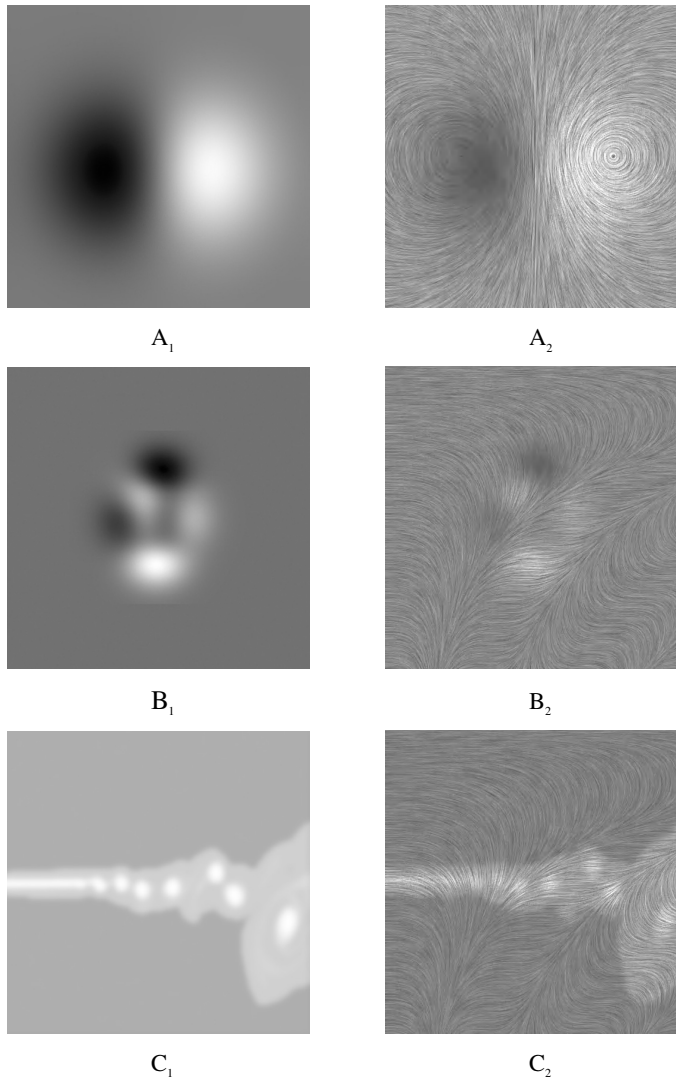


Fig. 4. Examples (A_1 B_1 C_1 is scalar values)

5 Conclusion

This paper presents a new and robust algorithm to tackle the multivariate visualization problem. Due to the sensitivity of the human eye to different contrast levels, It is a effective approach to map scalar values to the image contrast. Considering that LIC calculation is a time consuming task, we modifies the formula to do some simplification. With the development of the modern graphic card, we can make use of the programmable function of the graphic card to accelerate our algorithm to greatly

decrease the calculation time. The future work will be aimed to combine multi technique such as color mapping, tone mapping and contrast mapping together, make full use of graphic hardware. and offer better visualization result to the users.

Acknowledgements. This work is co-supported by the Education Ministry Excellent Young Teacher Awarding Funds, 973 project(grant no:2002G3312100); Zhejiang Province Talent Funds (RC40008).

References

[1] Jarke J Van Wijk. Spot Noise: Texture Synthesis For Data Visualization [J]. Computer Graphics, 1991, 25(4): 309–318.
[2] Brian Cabral, Leith Leedom. Imaging Vector Fields Using Line Integral Convolution [J]. Computer Graphics, 1993, 27(4): 263–270.
[3] Detlev Stalling, Hans Christian Hege. Fast and Resolution- Independent Line Integral Convolution. Proceedings of SIGGRAPH'95 [C]. 1995, 249–256.
[4] Hans-Christian Hege, Detlev Stalling. Fast LIC with Piecewise Polynomial Filter Kernels. Mathematical Visualization Algorithms and Applications [C]. Springer-Verlag, 1998, 295–314
[5] Lisa K Forssell. Visualizing Flow Over Curvilinear Grid Surfaces Using Line Integral Convolution[C]. Proceedings of IEEE Visualization'94:240–247,1994.2
[6] Han-Wei Shen, David L Kao. UFLIC: A Line Integral Convolution Algorithm For Visualizing Unsteady Flows. Proceedings of IEEE Visualization 97 [C]. 1997, 317–322
[7] H.W Shen, C.R. Johnson and K.L. Ma. Visualizing Vector Fields Using Line Integral Convolution and Dye Advection[C]. Symposium on Volume Visualization'96:63–70 1996.2
[8] H.W. Shen. Using Line Integral Convolution to Visualize Dense Vector Fields[J]. Computer in Physics,11(5):474–478.1997.2
[9] Sanna, B. Montrucchio. Adding a scalar to 2D vector field visualization: the BLIC (Bumped LIC)[C]. Eurographics' 2000 Short Presentations Proceedings:119–124,2000. 2,2,2
[10] A.Sanna, C.Zunino, B.Montrucchio and P.Montuschi. Adding a scalar value to texture-based vector field representations by local contrast analysis[C]. IEEE TCVG Symposium on Data Visualization(2002):35–41,2002.2
[11] Han-Wei Shen and David L. Kao A New Line Integral Convolution Algorithm for Visualizing Time-Varying Flow Fields[C]. IEEE transaction on visualization and computer graphics, Vol. 4, No. 2, April-June 1998
[12] Zhanping Liu, Guoping Wang, Shihai Dong A New Method of VolumeLIC for 3D Vector Field Visualization[J] Journal of Image and Graphics, 2001.5 Vol.6 No.5, pp. 47–474
[13] Zhang Wen, Li Xiao-mei 2D Vector Field Visualization Based on Streamline Texture Synthesis: [J] Journal of Image and Graphics, 2001.3 Vol.6 No.3, pp. 280–284
[14] Zhigeng Pan, Jiaoying Shi, Mingmin Zhang: Distributed graphics support for virtual environments. Computers & Graphics 20(2): 191–197 (1996)

Network Probabilistic Connectivity: Exact Calculation with Use of Chains*

Olga K. Rodionova[1], Alexey S. Rodionov[1], and Hyunseung Choo[2]

[1] Institute of Computational Mathematics and Mathematical Geophysics
Siberian Division of the Russian Academy of Science
Novosibirsk, RUSSIA +383-2-396211
alrod@rav.sscc.ru
[2] School of Information and Communication Engineering
Sungkyunkwan University
440-746, Suwon, KOREA +82-31-290-7145
choo@ece.skku.ac.kr

Abstract. The algorithmic techniques which allow high efficiency in the precise calculation of the reliability of an undirected graph with absolutely reliable nodes and unreliable edges are considered in this paper. The new variant of the branching algorithm that allow branching by chains is presented along with improvement of series-parallel reduction method that permits the reduction of a chain with more than two edges by one step. Some programming techniques which accomplish high efficiency are also discussed. Special notice is given to the problem of computer storage economy. Comprehensive computer simulation results show the advantages of the proposed algorithms, that the calculation time decreases significantly in comparison with existent methods.

1 Introduction

The task of calculating or estimating the probability of whether the network is connected (often referred to as its *reliability*, is the subject of much research due to its significance in a lot of applications, communication networks included. The problem is known to be NP-hard irrelative of whether the unreliable edges or nodes or both are considered. Most explored is the case of absolutely reliable nodes and unreliable edges that corresponds to real networks in which the reliability of nodes is much higher than that of edges. The transport and radio networks are good examples. Usually the estimations of a network reliability are considered. Yet by taking into consideration some special features of real network structures and based on modern high-speed computers we can conduct the exact calculation of reliability for networks with dimension of a practical interest.

* This paper was partially supported by BK21 program, University ITRC and RFBR. Dr. Choo is the corresponding author.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 315–324, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The well-known branching algorithm (often called Moore-Shannon algorithm [1]) uses branching on the alternative states for an arbitrary edge. Our first approach is to branch by the whole chain if it exists. Another well-known approach tWe compare our algorithms to previous works including the technique proposed in [6] for performance evaluation, and those proposed in this paper are much faster.hat uses series-parallel reduction owes to its spreading mostly to A.M. Shooman [2,3]. However in the reduction of series this method uses consequent reduction of pairs of edges. We propose to reduce the entire chain at once thereby allowing the algorithm to improve the speed in cases with networks with long chains. The proper theorems are proven in this paper to support the proposed methods. The programming of the proposed algorithms is non-trivial. In this paper we are trying give a proper attention to this task. Special notice is given to the problem of computer storage economy.

The rest of the paper is organized as follows: in section 2 you can find the derivation of the modified branching and series-parallel reduction methods and the technique for preliminary lowering of a problem dimension. In section 3 the computer algorithm is presented. Rules of break and results of various kinds of the graph contracting by a chain or its deletion are considered. Section 4 contains the discussion of computational experiments. It is shown through experimentation that our approaches allow exact calculation of the reliability of networks with practically interested dimensions. Section 5 is the brief conclusion.

2 Using Chains in the Calculation of Network Reliability

As the treating of dangling nodes, articulation nodes and bridges in the reliability calculation is well-known we consider the initial network structures that are free of them.

The branching method mentioned above is the most widely known (often by the name "factoring method" also) for exact calculation of a graph reliability. Its formula is:

$$R(G) = p_{ij}R(G^*(e_{ij})) + (1 - p_{ij})R(G\backslash\{e_{ij}\}), \tag{1}$$

where $G^*(e_{ij})$ – graph contracted by an edge e_{ij} that exists with probability p_{ij}, $G\backslash\{e_{ij}\}$ – graph obtained from G by deletion of the edge e_{ij}. The recursions go on till deriving the disconnected graph (returns 0), or the graph of small dimension (2, 3 or 4 nodes) for which the reliability is easily obtained. In [4, 5] the modification of the branching method is presented which permits the branching by chains of edges transiting through nodes with degree 2.

Theorem 1. *Let a graph G have a simple chain $Ch = e_1, e_2, \ldots, e_k$ with edge reliabilities $p_1, p_2, \ldots, p_k$, respectively, connecting nodes s and t. Then the reliability of G is equal to*

$$R(G) = \prod_{j=1}^{k} p_j \cdot R(G^*(Ch)) + \sum_{i=1}^{k}(1 - p_i)\prod_{j\neq i} p_j \cdot R(G\backslash Ch), \tag{2}$$

if e_{st} does not exist and

$$R(G) = \left[(p_1 + p_{st} - p_1 p_{st}) \prod_{j=2}^{k} p_j + p_{st} \sum_{i=2}^{k} (1 - p_i) \prod_{j \neq i} p_j \right] \times R(G^*(Ch)) + \quad (3)$$

$$\left[(1 - p_1)(1 - p_{st}) \prod_{j=2}^{k} p_j + (1 - p_{st}) \sum_{i=2}^{k} (1 - p_i) \prod_{j \neq i} p_j \right] \times R(G \backslash Ch \backslash e_{st}),$$

otherwise, where $G^*(Ch)$ is a graph obtained from G by contracting by the chain, $G\backslash Ch$ is a graph obtained from G by deletion of this chain with nodes (except for terminal ones), and p_{st} is the reliability of an edge directly connecting the terminal nodes of the chain.

The **proof** of this theorem is obtained by applying the formula (1) to the edges $e_k, e_{k-1}, \ldots, e_1$ consequently. The backtracking process of sum the reliabilities of terminal graphs multiplied by the probabilities of branches gives us the proof of the theorem □.

Let us note, that even if there are no such chains in the initial graph, they can appear during the recursive branching. Later on we will refer to the chain by which the branching is done as a *resolving chain*. We always start with the deleting as it is possible to obtain a disconnected graph by it. In this case we know that the resolving chain is a bridge and we can obtain the reliability of our graph by multiplication of reliabilies of two graphs of smaller dimension and the reliability of this bridge.

A.M. Shooman [2,3] has proposed substituting the parallel or subsequent pair of edges to one to speed up the reliability calculation. Thus the graph G is transformed to some graph G^* with smaller number of edges and, possibly, nodes. Reducing k parallel edges is obvious and simple

$$p = 1 - \prod_{i=1}^{k} (1 - p_i), \quad (4)$$

while the reducing of an consequent pair of edges leads to a graphs with a different reliability:

$$R(G) = rR(G^*), \quad (5)$$

$$p = \frac{p_1 p_2}{1 - (1 - p_1)(1 - p_2)} = \frac{p_1 p_2}{p_1 + p_2 - p_1 p_2}, \quad (6)$$

$$r = p_1 + p_2 - p_1 p_2. \quad (7)$$

Based on this result and the consequent reduction on pairs of edges for the chain with length $k > 2$ we can formulate the following theorem [5].

Theorem 2. *Let a graph $G_1(n, m)$ have a simple chain $Ch = e_1, e_2, \ldots, e_k$ with edge reliabilities $p_1, p_2, \ldots, p_k$, respectively, connecting nodes s and t. Then*

$$R(G_1(n, m)) = \prod_{i=1}^{k} p_i \left(\sum_{i=1}^{k} {p_i}^{-1} - k + 1 \right) R(G_2(n - k + 1, m - k + 1)), \quad (8)$$

where a graph $G_2(n-k+1, m-k+1)$ *is derived from* $G_1(n,m)$ *by substituting the chain by a single edge with the probability of the edge existence*

$$p = 1/\left(\sum_{i=1}^{k} p_i^{-1} - k + 1\right). \tag{9}$$

Proof of the theorem is based on the mathematical induction. □

After substituting all chains by edges the reduced graph is calculated by the simple branching method. If during the process a new chain appears, then it is also substituted by an edge. Reducing all chains with consequent branching is faster than branching by chains as it leads to small-dimension graphs on earlier recursions (see Fig. 1). Further consideration of the example is made in the section 4.

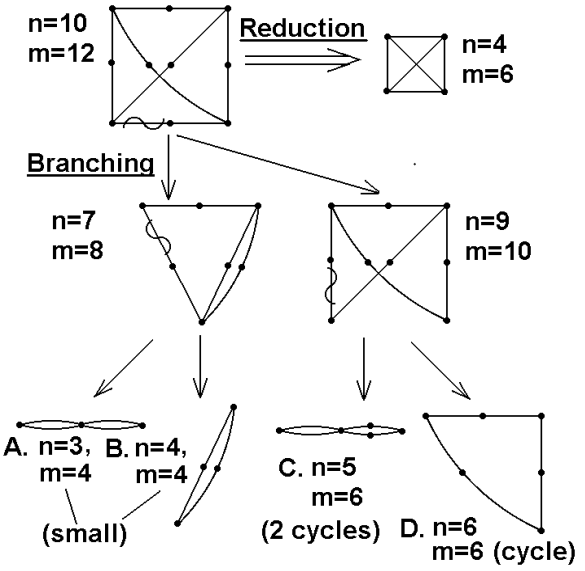


Fig. 1. Comparison of chains' reduction and branching by chain

3 Program Realization of the Algorithms

The problem of programming the proposed algorithms is not trivial by virtue of the high request to the memory, and of numerous recursions also. We discuss the following aspects in this section: (1) re-usage of memory in recursions; (2) finding chains for branching and reduction; (3) renumbering nodes; and (4) the final graphs that allow direct calculation. Because of the limited paper size we discuss the first three items here while the final graphs are just listed with some comments.

Memory Re-usage. Let us consider the the branching process. If we use (1) or (2) and (4) then we have 2 recursive calls of the base method *with different graphs*. The main part of the input data is the presentation of the corresponding graph. The probabilistic matrix $P = \|p_{ij}\|$ is most convenient for this task. The production of *new* probabilistic matrices is ineffective as it can lead to overloading the memory. Therefore essential in realization of these algorithms is re-usage of the probabilistic matrix of a graph. It must be prepared for input on the next recursion at branching and restored after exiting from it.

In the simple branching we choose an edge for branching among those that are connected with the last node (say, n-th). Thus the preparation for contracting is recalculating the values of probabilities p_{im} and $p_{mi},\ i = 1, \ldots, n-1$, where m is the second node incident to the edge chosen, and reducing the dimension by 1 (left-upper block of the matrix with $n-1$ rows and columns). The old values are stored (remember that $p_{im} = p_{mi}$). The preparation for deleting is trivial: p_{nm} and p_{mn} are zeroed (the old value is stored). After returning from the recursion the process revolves.

In the branching by chains the task is harder: the chain can go through nodes with arbitrary numbers so first we need to renumber them in such a way that the deleted nodes are with last numbers. The task of renumbering is discussed later, the correspondence "old numbers – new numbers" must be stored also. In this case the probability matrix for recursive call is prepared and restored almost as simple as in the previous case. The difference is in dimension: by contracting it is $n - k$ and by deleting – $n - k + 1$ nodes where k is the chain length.

Finding the Resolving Chain. There is a desire to use the longest simple chain of the graph as resolving. However that requires determination of all chains and comparison of their lengths. Therefore a chain which includes a node of a degree 2 with minimum node number (let this node be v_{k_0}) is simply searched. The list of nodes is constructed in two directions starting from v_{k_0} till the terminal nodes have degree 2 also.

Let the resolving chain be (a list of consequent nodes):

$$Ch = (v_{k_{-s}}, \ldots, v_{k_{-1}}, v_{k_0}, \ldots, v_{k_t}) \tag{10}$$

As $N(H)$ we will denote the set of numbers of nodes that belong to some subgraph H of the graph G. Thus $N(Ch) = \{k_{-s}, k_{-s+1}, \ldots, k_0, k_1, \ldots, k_t\}$ and $N(G) = \{1, \ldots, n\}$.

We check the minimum degree dynamically using constantly updated array Deg of node degrees.

Renumbering Nodes in a Resolving Chain. Renumbering of the nodes is needed not only in branching by chains but also at chain reduction. The rule of renumbering is the same for both proposed algorithms so later we consider only the case of branching.

The chain should be contracted to a node with node number $n-k$ (dimension of the reduced graph), thus this number is assigned to one of two terminal nodes. The number $n - k + 1$ is assigned to the other one, that ensures conformity of

the deletion of the resolving chain to the simple reduction of the probability matrix dimension. Thus the numbers of nodes of the resolving chain (including terminal) should be $n-d, n-d+1, \ldots, n$ after renumbering, where d is the number of edges for the chain, and n is the number of nodes for the graph under reduction. We need to make the following change for node numbers (old numbers are labeled as in (10)):

$$k_{-s} \longrightarrow n-d,\ k_t \longrightarrow n-d+1,\ k_i \longrightarrow n-d+s+i+1,\ i=-s+1,\ldots,t-1. \quad (11)$$

It is possible that $\exists i : (i \in \{n-d, n-d+1, \ldots, n\}) \wedge (i \in N(G) \backslash N(Ch))$. Let us denote the set of such numbers as S_{add}. For each node $v_i | i \in S_{add}$ we assign the new number from the set $U = N(Ch) \backslash \{n-d, n-d+1, \ldots, n\}$. The natural way is to arrange numbers in S_{add} in ascending order and choose the correspondent new numbers from U in the same fashion. Thus we obtain two lists of old and new numbers for some subset of nodes of the graph needed for the renumbering procedure:

$$N_{old} = N(Ch) \bigcup S_{add}, \quad (12)$$
$$N_{new} = \{n-d, n-d+1, \ldots, n\} \bigcup U.$$

The examples of renumbering are presented in Fig. 2. Here the new numbering for the nodes are indicated in parentheses. Our programming of the algorithm for the renumbering is conducted with the usage of intermediate presentation of the graph by its list of edges, that is the set of pairs of node numbers. At the sequential consideration for these pairs node numbers in pairs change from old to new (if these numbers belong to the set of those changing). The new probability matrix is then constructed. For this we need the intermediate vector of edge reliabilities. At the same time, for the renumbering of a pair of nodes which is required at the deletion of a dangling node or reduction of a chain with the length of 2, the usage of intermediate presentation is not necessary.

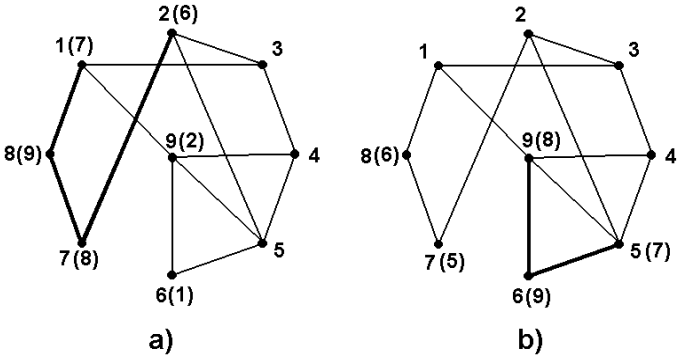


Fig. 2. Examples of the node's renumbering at a different choices of the resolving chain

Variants of Results on Contracting and Removal Operations. On execution of branching it is necessary to take into account all possible variants of the resulting graphs. While performing the classical branching method (1) there are only 3 possible results: the derivation of a disconnected graph at deletion of an edge, a graph of small dimension simple for calculation at contracting and a graph that is connected but not possible for direct calculation yet, to which the operation of branching is applied again. At usage of the formulas (2) and (4) in the branching by chain it is necessary to take more variants into account.

In our programming we have found the following variants that demand special treating, first three are obvious: (1) the resulting graph is a cycle; (2) the resolving chain is a cycle; (3) the dangling node appears; (4) the resulting graph is disconnected. The last means that any edge in the deleted chain is a bridge. Accordingly, by contracting we obtain a articulation point and the reliability of the graph is considered as the product of the reliabilities of two graphs G_1 and G_2 and probability of the existence of a resolving chain (or edge).

Note that at contracting it is possible to obtain a multi-edge (not more than 2 parallel edges) that must be changed to an edge with equivalent reliability.

The Final Graphs with Small Dimension. The simplest graph has one edge and two nodes. In this case the reliability of the edge is returned. However it would be desirable to calculate the reliability directly for graphs as large as possible, since it saves us from the necessity of further recursive procedure calls. Also, because of the plural repeated execution of the calculation formula, it is desirable to construct it optimally. Thus, for a case of three nodes (reliabilities of edges are a, b and c) we have

$$R(G) = abc + ab(1-c) + a(1-b)c + (1-a)bc, \text{ or}$$
$$d = ab, \quad R(G) = c(a+b) + d - 2cd.$$

The first variant takes 8 operations of multiplication and 6 of addition/substraction while the second variant takes only 4 operations of multiplication and 3 of addition/substraction. In our realization of the algorithm we use the formulas for computing the reliability of 4-node complete graph that by the application of the intermediate variables allows implementation of the calculations by 28 operations of multiplication and 31 operations of addition/substraction.

Note, that after contracting the graph with number of nodes more than 4 it is possible to obtain a graph that has not only 4 or 3, but even 2 nodes. Therefore it is necessary to check up for all these variants. However, this situation never occurs when we use chain reduction before contracting.

On the basis of the material explained above we propose the algorithm for the recursive procedure for strict calculation of the reliability of a graph without multiple edges. This algorithm assumes reduction of dangling nodes and chains in the input graph and branching only in the case of their absence. All possible variants of final graphs are checked and cycles and articulation points are treated as was discussed.

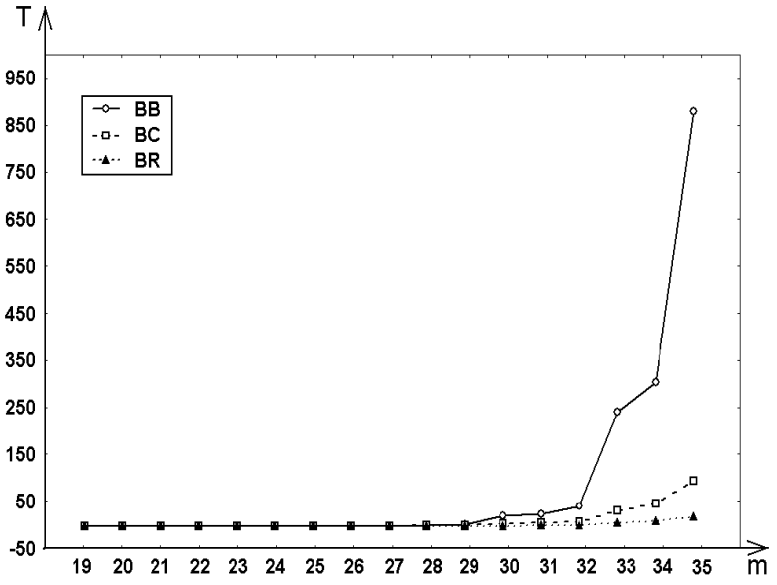


Fig. 3. Dependence of calculation time spent on 30 random 20-node graphs on number of edges

4 Case Studies

We conducted several experiments on the computer with the processor AMD Athlon 800MHz inside. We have made the comparisons among the algorithm with branching by chains (BC), basic branching algorithm (BB), branching algorithm with chain reduction (BR) and algorithm from [6] (RT).

In the example of the lattice (4×4) graph, that was used in [6], the number of basic recursion for RT is 2579141, time spent for calculation was about 47 seconds. Algorithm BC takes 0.17 seconds and only 407 recursions on this example. Note, that 200 chains were found during the calculation with average length 2.385. So on this example our algorithm is more than 200 times faster. The basic BB algorithm takes on this example 8.35 seconds, which is about 50 times slower than BC and takes 80619 recursions. However best results were shown by the BR algorithm which takes only 0.06 seconds on 93 recursions. When the dimension of a lattice was increased up to (5×5) the algorithm RT did not finished in 2 hours, and BB, BC and BR algorithms took 21 minutes, 15.05 and 2.47 seconds on 13817311, 51652 and 14581 recourses respectively.

Another example is calculating the probability of the graph in Fig. 1. The basic BB algorithm takes on this example 0.06 seconds and 139 recursions. At the same time both BC and BR take less than 0.1 seconds and 16 recursion on BC and none on BR. Note that in Fig. 1 we choose resolving chains optimally while the program makes branching by the first found chain. Worse is the algorithm

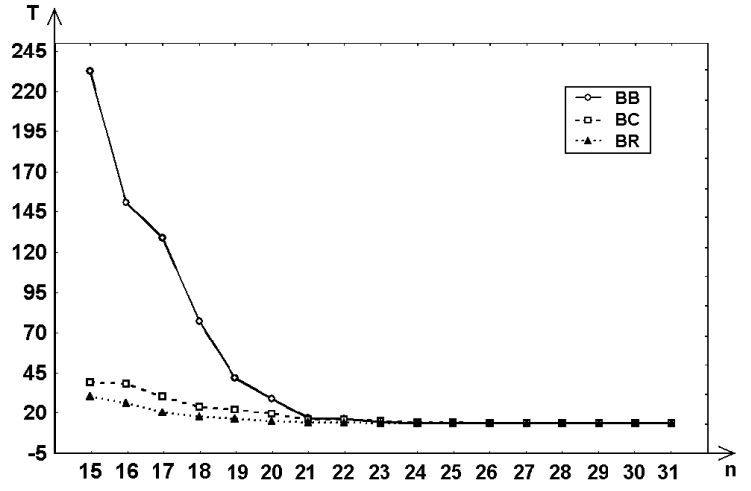


Fig. 4. Dependence of calculation time spent on 30 random 30-edge graphs on number of nodes

RT that takes 0.28 seconds on 112 recursions. Thus our algorithms shows better efficiency again.

In the figure 3, the dependence of calculation time on the number of edges is shown for the examined algorithms for 20-node graphs with number of edges from 19 to 35 (total for 30 random graphs for each case). In the figure 4 for examined algorithms the dependence of calculation time for the number of nodes is shown for a 30-edge graph with the number of nodes from 15 to 31 (total for 30 random graphs for each case). From the results the advantage of the algorithm BR is clear. In all examined methods for the graphs with near numbers of nodes and edges the calculation time is very small and almost the same. Obviously this time is primarily the time spent for random graph generation and the output of results.

Last we calculate the reliability of the graph with the structure of well-known ARPA network. This graph has 58 nodes and 71 edges. The algorithm BC takes approximately 20 minutes and BR – about one minute for calculation. With this the last algorithm takes only 31933 recursions.

5 Conclusion

In this paper we have shown how to use chains for speeding up the process of obtaining the exact reliability of networks with reliable nodes and unreliable edges. Although the idea of chain reduction is not new and is well explored by A.M. Shooman we are first who propose to reduce a long chain by one step. The branching by chains is completely new idea and being less efficient than the main algorithm presented in the paper is still much more effective than previous algorithms and is easier for programming. The thorough experiments

show that our algorithms can be used for strict calculation of the reliability of networks with dimensions of practical interest. We think that our method can be used for topological optimization of networks using the method proposed in [10] in which the calculation of a graph reliability is one of the main subgoals. Future researches can concern the exact calculation of reliability for networks with unreliable nodes also.

References

1. Moore, E.F., Shannon, C.E., "Reliable Circuits Using Less Reliable Relays," *J. Franclin Inst., 262, n. 4b,* pp. 191–208, 1956.
2. Shooman, A.M., Kershenbaum, A., "Exact Graph-Reduction Algorithms for Network Reliability Analysis," *Proc. GLOBECOM' 91. Vol. 2*, pp. 1412–1420, 1991.
3. Shooman, A.M., "Algorithms for Network Reliability and Connection Availability Analysis," *Electro/95 Int. Professional Program Proc.,* pp. 309–333, 1995.
4. Rodionov, A.S., Rodionova, O.K., "On a Problem of Practical Usage of the Moore-Shennon Formula for Calculating the Reliability of Local Networks," *Proc. 2nd Int. Workshop INFORADIO-2000*, Omsk, pp. 67–69, 2000.
5. Rodionova, O.K., "Some Methods for Speed up the Calculation of Information Networks Reliability," *Proc. XXX International Conf. "IT in Science, Education, Telecommunications and Business," Ukraine, Gurzuf,* pp. 215–217, 2003.
6. Chen, Y., Li, J. Chen, J., "A new Algorithm for Network Probabilistic Connectivity," *Proc. MILCOM'99. IEEE, Vol. 2,* pp. 920–923, 1999.
7. Rodionova, O.K. "Application Package GRAPH-ES/3. Connectivity of the Multigraphs with Unreliable Edges (Atlas, procedures)," Preprint No. 356, Computing Center of the SB AS of the USSR, Novosibirsk, 1982. (in Russian)
8. Rodionova, O.K., Gertzeva, A.A., "On the Construction of thr Optimal-connected graphs," *Proc. of the ICS-NET'2001 Int. Workshop, Moscow,* pp. 200–204, 2001. (in Russian)
9. Tolchan, A.Y. "On the Network Connectivity," *Problems of the information transmission, Issue 17,* 1964, pp. 3–7. (in Russian)
10. T. Koide, S. Shinmori and H. Ishii, "Topological optimization with a network reliability constraint," *Discrete Appl. Math., vol. 115, Issues 1-3,* pp. 135–149, November 2001.

Curvature Dependent Polygonization by the Edge Spinning

Martin Čermák* and Václav Skala

University of West Bohemia, Pilsen
Department of Computer Science and Engineering
Czech Republic
{cermakm,skala}@kiv.zcu.cz

Abstract. An adaptive method for polygonization of implicit surfaces is presented. The method insists on the shape of triangles and the accuracy of resulting approximation as well. The presented algorithm is based on the surface tracking scheme and it is compared with the other algorithms based on the similar principle, such as the Marching cubes and the Marching triangles methods. The main advantages of the triangulation presented are simplicity and the stable features that can be used for next expanding.

1 Introduction

Implicit surfaces seem to be one of the most appealing concepts for building complex shapes and surfaces. They have become widely used in several applications in computer graphics and visualization.

An implicit surface is mathematically defined as a set of points in space **x** that satisfy the equation $f(\mathbf{x}) = 0$. There are two different definitions of implicit surfaces. The first one [2], [3] defines an implicit object as $f(\mathbf{x}) < 0$ and the second one, F-rep [9], [11], [12], defines it as $f(x) \geq 0$.

Existing polygonization techniques may be classified into three categories. Spatial sampling techniques that regularly or adaptively sample the space to find the cells that straddle the implicit surface [2], [4]. Surface tracking approaches iteratively create a triangulation from a seed element by marching along the surface [1], [2], [5], [7], [10], [16]. Surface fitting techniques [11] progressively adapt and deform an initial mesh to converge to the implicit surface.

2 Algorithm Overview

Our algorithm is based on the surface tracking scheme (also known as the continuation scheme) and therefore, there are several limitations. A starting point must be determined and only one separated implicit surface is polygonized for such

* This work was supported by the Ministry of Education of the Czech Republic – project MSM 235200002.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 325–334, 2004.
© Springer-Verlag Berlin Heidelberg 2004

point. Several disjoint surfaces can be polygonized from a starting point for each of them.

The algorithm uses only the standard data structures used in computer graphics. The main data structure is an edge that is used as a basic building block for polygonization. If a triangle's edge lies on the triangulation border, it is contained in the *active edges list* (*AEL*) and it is called as an *active edge*. Each point, which is contained in an active edge, contains two pointers to its left and right active edge (left and right directions are in active edges' orientation). The whole algorithm consists of the following steps:

1. Initialize the polygonization:
    a. Find the starting point $\mathbf{p}_0$ and create the first triangle T_0., see [5] for details.
    b. Include the edges (e_0,e_1,e_2,) of the first triangle T_0 into the active edges list.
2. Polygonize the first active edge *e* from the active edges list.
3. Update the *AEL*; delete the currently polygonized active edge *e* and include the new generated active edge/s at the end of the list.
4. If the active edges list is not empty return to step 2.

3 Edge Spinning

The main goal of this work is a numerical stability of a surface point coordinates' computation for objects defined by implicit functions. In general, a surface vertex position is searched in direction of a gradient vector $\nabla\mathbf{f}$ of an implicit function *f*, as in [7]. In many cases, the computation of gradient of the function *f* is influenced by a major error that depends on modeling techniques used [9], [10], [11], [12], [14], [15]. Because of these reasons, in our approach, we have defined these restrictions for finding a new surface point $\mathbf{p}_{new}$:

- The new point $\mathbf{p}_{new}$ is sought on a circle; therefore, each new generated triangle preserves the desired accuracy of polygonization. The circle radius is proportional to the estimated surface curvature.
- The circle lies in the plane that is defined by the normal vector of triangle T_{old} and axis *o* of the current edge *e*, see Fig. 2; this guarantees that the new generated triangle is well shaped (isosceles).

3.1 Circle Radius Estimation

The circle radius is proportional to the estimated surface curvature. The surface curvature in front of current active edge is determined in according to angle α between the surface normals $\mathbf{n}_1$, $\mathbf{n}_2$, see Fig. 1. The normal vector $\mathbf{n}_1$ is computed at point **s** that lies in the middle of the current active edge *e* and the vector $\mathbf{n}_2$ is taken at initial point $\mathbf{p}_{init}$ that is a point of intersection of the circle c_1 with the plane defined by the triangle T_{old}.

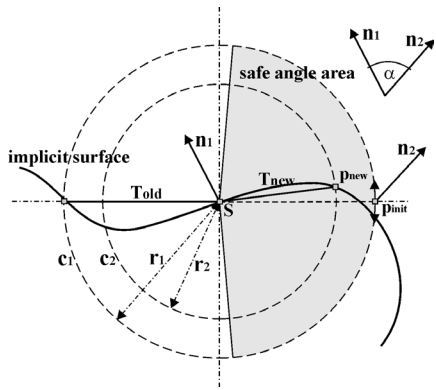


Fig. 1. The circle radius estimation.

Note that the initial radius r_1 of the circle c_1 is always the same and it is set at beginning of polygonization as the lowest desired level of detail (LOD).
The new circle radius r_2 is computed as follows.

$$r_2 = r_1 \cdot k, k \in \langle 0,1 \rangle; \qquad (1)$$
$$k = \left(\frac{\alpha_{\lim} - \alpha \cdot c}{\alpha_{\lim}} \right),$$

where α_{lim} is a limit angle and the constant *c* represents a speed of "shrinking" of the radius according to the angle α. To preserve well shaped triangles, we use a constant k_{min} that represents a minimal multiplier. In our implementation we used $\alpha_{min} = \pi/2$, $k_{min} = 0.2$ and $c = 1.2$.

Correction notes:

if $(\alpha > \alpha_{min})$ then $k = k_{min}$
if $(k < k_{min})$ then $k = k_{min}$

These parameters affect a shape of triangles of the polygonal mesh generated.

3.2 Root Finding

If the algorithm knows the circle radius, the process continues as follows.

1. Set the point $\mathbf{p}_{new}$ to its initial position; the initial position is on the triangle's T_{old} plane on the other side of the edge *e*, see Fig. 2. Let the angle of the initial position be $\alpha=0$.
2. Compute the function values $f(\mathbf{p}_{new}) = f(\alpha)$,
   $f(\mathbf{p'}_{new}) = f(\alpha + \Delta\alpha)$ – initial position rotated by the angle $+\Delta\alpha$,
   $f(p''_{new}) = f(\alpha - \Delta\alpha)$ - initial position rotated by the angle $-\Delta\alpha$; Note that the rotation axis is the edge *e*.
3. Determine the right direction of rotation; if $|f(\alpha + \Delta\alpha)| < |f(\alpha)|$ then $+\Delta\alpha$ else $-\Delta\alpha$.
4. Let the function values $f_1 = f(\alpha)$ and $f_2 = f(\alpha \pm \Delta\alpha)$; update the angle $\alpha = \alpha \pm \Delta\alpha$.

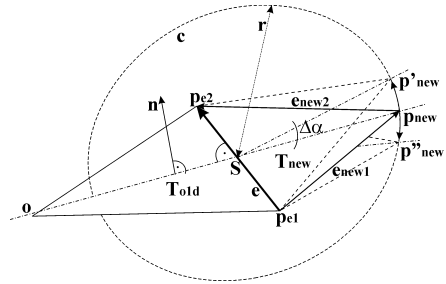


Fig. 2. The principle of root finding algorithm.

5. Check which of following case appeared:

a) If $(f_1 \cdot f_2)<0$ then compute the accurate coordinates of the new point $\mathbf{p}_{new}$ by the binary subdivision between the last two points which correspond to the function values f_1 and f_2;

b) If the angle $|\alpha|$ is less than α_{safe} (see *safe angle area* in Fig. 1) return to step 4.

c) If the angle $|\alpha|$ is greater than α_{safe} then there is a possibility that both triangles T_{old} and T_{new} could cross each other; the point $\mathbf{p}_{new}$ is rejected and it is marked as *not found*.

3.3 Root Finding of a Sharp Edge

Let us assume that the standard edge spinning root finding algorithm presented above has found the point $\mathbf{p}_{new}$. The algorithm then determines the surface normal vector $\mathbf{n}_{new}$ at this point and computes the angle α between normal vectors $\mathbf{n}_{new}$ and $\mathbf{n}_s$. The vector $\mathbf{n}_s$ is measured at mid-point **s** of the active edge *e*, see Fig. 3. If the angle α is greater then some user-specified threshold α_{lim_edge} (limit edge angle) then the algorithm will look for a new edge point as follows.

1. Compute coordinates of the point $\mathbf{p}_{init}$ as an intersection of the three planes, tangent planes $\mathbf{t}_1$ and $\mathbf{t}_2$, and the plane in which the seeking circle *c* lies, see Fig. 3.
2. Apply the straight root finding algorithm described in section 3.4 and find the new point $\mathbf{p}'_{new}$.

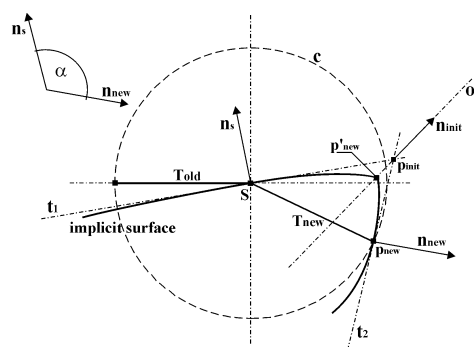


Fig. 3. The principle of root finding algorithm for sharp edges.

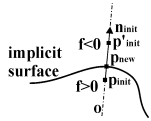


Fig. 4. Principle of root-finding in straight direction.

3.4 Straight Root Finding Algorithm

The algorithm starts from an initial point $\mathbf{p}_{init}$ (see Fig. 4) and supposes that the implicit surface is at least C^0 continuity.

1. At point $\mathbf{p}_{init}$, compute the surface normal vector $\mathbf{n}_{init}$ that defines the seeking axis *o*.
2. Compute coordinates of point $\mathbf{p}'_{init}$ with distance δ from point $\mathbf{p}_{init}$ in direction $\mathbf{n}_{init} * \text{sign}(f(\mathbf{p}_{init}))$; where δ is the length of step and the function sign returns "1" if ($f > 0$) or "0" if ($f < 0$).
3. Determine function values f, f' at points $\mathbf{p}_{init}$, $\mathbf{p}'_{init}$.
4. Check next two cases.
   a. If these points lie on opposite sides of implicit surface, i.e. $(f * f') < 0$; compute the exact coordinates of the point $\mathbf{p}_{new}$ by binary subdivision between these points.
   b. If the points $\mathbf{p}_{init}$, $\mathbf{p}'_{init}$ lie on the same side of the surface then $\mathbf{p}_{init} = \mathbf{p}'_{init}$ and return to step 2.

4 Polygonization of an Active Edge

Polygonization of an active edge *e* consists of several steps. In step 1, the process will use the root finding algorithm (see section 3.2) to find a new point $\mathbf{p}_{new}$ in front of the edge *e*. If $\mathbf{p}_{new}$ exists, there are two cases illustrated in Fig. 5.

4.1 Neighborhood Test

Decision between cases a) and b) depends on relation among angles α_1, α_2, α_n, see Fig. 5, step 1; let the angle α be $min(\alpha_1,\alpha_2)$. If ($\alpha < \alpha_{shape}$) then case a) else case b), see Fig. 5, step 2; The limit shape angle is determined as $\alpha_{shape} = k*\alpha_n$, $k \geq 1$, $\alpha_{shape} < \pi$, where the constant *k* has effect to shape of generated triangles and in our implementation is chosen $k = 1.7$. If the point $\mathbf{p}_{new}$ is not found, angle α_n is not defined and the limit shape angle should be just less then π; we have chosen $\alpha_{shape} = \pi*0.8$.

a) In this case, a new triangle t_{new} is created by connecting the edge *e* with one of its neighbors, see step 2a.
b) The new triangle t_{new} is created by joining the active edge *e* and the new point $\mathbf{p}_{new}$, see step 2b.

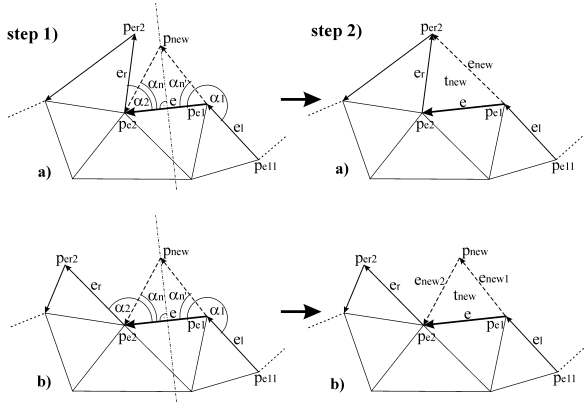


Fig. 5. Polygonization of the active edge *e*.

In both cases, a bounding sphere is determined for the new triangle t_{new}. The bounding sphere is the minimal sphere that contains all three points of the triangle, i.e. the centre of the sphere lies in the plane defined by these three points. If there is not a new triangle (the point $\mathbf{p}_{new}$ does not exist and case a) has not appeared) the bounding sphere of the active edge *e* is used. The next procedure is analogical for all cases.

4.2 Distance Test

To preserve the correct topology, it is necessary to check each new generated triangle if it does not cross any other triangles generated before. It is sufficient to perform this test between the new triangle and a border of already triangulated area (i.e. active edges in *AEL*). For faster evaluation of detection of global overlap there is used the space subdivision acceleration technique introduced in [6].

The algorithm will make the *nearest active edges list* (*NAEL*) to the new triangle t_{new}. Each active edge that is not adjacent to the current active edge *e* and crosses the bounding sphere of the new triangle (or the edge *e*), is included to the list, see Fig. 6, step 2. The extended bounding sphere is used for the new triangle created by the new point $\mathbf{p}_{new}$ (case b) because the algorithm should detect a collision in order to preserve well-shaped triangles. The new radius of the bounding sphere is computed as $r_2 = c*r_1$ and we used the constant c = 1.5.

If the *NAEL* list is empty then the new triangle t_{new} is finally created and the active edges list is updated.

- In case a), Fig. 5 step 2, the current active edge *e* and its neighbor edge e_r are deleted from the list and one new edge e_{new} is added at the end of the list. The new edge should be tested if it satisfies the condition of the surface curvature. If it does not then the new triangle will be split along the edge e_{new}, see section 4.3.
- In case b) Fig. 5 step 2, the current active edge *e* is deleted from the list and two new edges e_{new1}, e_{new2} are added at the end of the list.

Note that if there is no new triangle to be created (the point $\mathbf{p}_{new}$ does not exist and case a) in Fig. 5 has not appeared) the current active edge *e* is moved at the end of the *AEL* list and the whole algorithm will return back to step 2, see section 2.

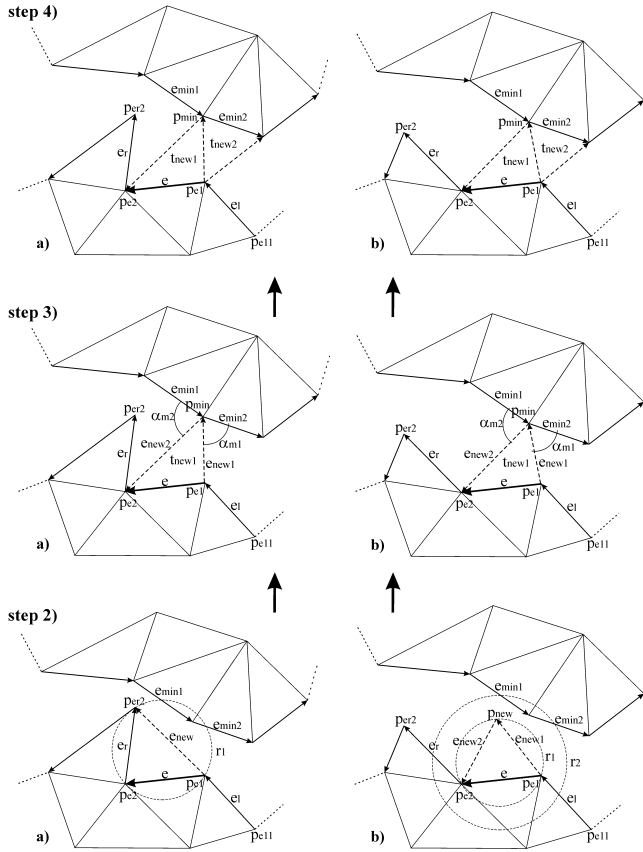


Fig. 6. Solving of distance test.

If the *NAEL* list is not empty then the situation has to be solved. The point $\mathbf{p}_{min}$ with minimal distance from the centre of the bounding sphere is chosen from the *NAEL* list, see Fig. 6, step 3. The new triangle t_{new} has to be changed and will be formed by the edge *e* and the point $\mathbf{p}_{min}$, i.e. by points ($\mathbf{p}_{e1}$,$\mathbf{p}_{min}$,$\mathbf{p}_{e2}$); the situation is described in Fig. 6, step 3. The point $\mathbf{p}_{min}$ is owned by four active edges e_{new1}, e_{new2}, e_{min1}, e_{min2} and the border of already triangulated area intersects itself on it. This is not correct because each point that lies on the triangulation border should has only two neighborhood edges (left and right).

Solution of the problem is to triangulate two of four edges first. Let the four active edges be divided into pairs; the left pair be (e_{min1}, e_{new2}) and the right pair be (e_{new1}, e_{min2}). One of these pairs will be polygonized and the second one will be cached in memory for later use. The solution depends on angles α_{m1}, α_{m2}, see Fig. 6, step 3. If ($\alpha_{m1} < \alpha_{m2}$) then the left pair is polygonized; else the right pair is polygonized.

In both cases, the recently polygonized pair is automatically removed from the list and the previously cached pair of edges is returned into the list. The point $\mathbf{p}_{min}$ is contained only in one pair of active edges and the border of the triangulated area is correct, Fig. 6, step 4.

Note that the polygonization of one pair of edges consists just of joining its end points by the edge and this second new triangle has to fulfill the empty *NAEL* list as well; otherwise the current active edge *e* is moved at the end of *AEL* list.

4.3 Splitting the New Triangle

This process is evaluated only in cases when the new triangle has been created by connecting of two adjacent edges, i.e. situation illustrated in Fig. 7, step 2a. If the new edge does not comply a condition of surface curvature the new triangle should be split. That means, see Fig. 7; if the angle α between surface normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$ at points $\mathbf{p}_{e1}$, $\mathbf{p}_{er2}$ is greater then some limit α_{split_lim} then the new triangle will be split into two new triangles, see Fig. 7, step 2.

The point $\mathbf{p}_{new}$ is a midpoint of edge e_{new} and it does not lie on the implicit surface. Its correct coordinates are additionally computed by the straight root finding algorithm described in section 3.4.

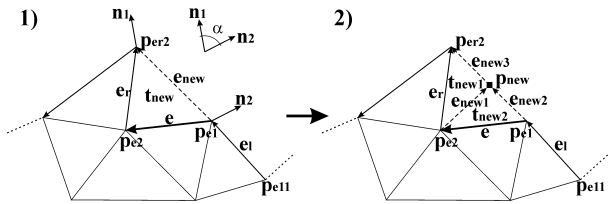


Fig. 7. Splitting of the new triangle.

5 Experimental Results

The Edge spinning algorithm (ES) is based on the surface tracking scheme (also known as the continuation scheme). Therefore, we have compared it with other methods based on the same principle – the Marching triangles algorithm (MTR, introduced in [7]) and the Marching cubes method (MC, Bloomenthal's polygonizer, introduced in [2]). As a testing function, we have chosen the implicit object Genus 3 that is defined as follows.

$$f(\mathbf{x}) = r_z^4 \cdot z^2 - \left[1 - (x/r_x)^2 - (y/r_y)^2\right] \cdot \left[(x - x_1)^2 + y^2 - r_1^2\right] \cdot \left[(x + x_1)^2 + y^2 - r_1^2\right] = 0$$

where the parameters are: $\mathbf{x} = [x,y,z]^T$, r_x=6, r_y=3.5, r_z=4, r_1=1.2, x_1=3.9.

The values in Table 1 have been achieved with the desired lowest level of detail (LOD) equal 0.8. It means that maximal length of triangles' edges is 0.8. Note that there is not defined a unit of length, so that number could be for example in centimeters as well as the parameters of the function Genus 3 described above.

The table contains the number of triangles and vertices generated. The value *Avg dev.* means the average deviation of each triangle from the real implicit surface. It is measured as algebraic distance of a gravity centre of a triangle from an implicit surface, i.e. the function value at the centre of gravity of the triangle. Note that the algebraic distance strongly depends on the concrete implicit function; in our test, the Genus 3 object is used for all methods, so the value has its usefulness.

Table 1. Values of the object Genus 3 with the lowest level of detail LOD = 0.8.

	ES	MTR	MC
# Triangles	4886	947	1056
# Vertices	2439	473	516
Avg dev.	10,99	56,80	73,28
Angle crit.	0,65	0,67	0,38
Elength crit.	0,77	0,78	0,54

The value *Angle crit*. means the criterion of the ratio of the smallest angle to the largest angle in a triangle and the value *Elength crit*. means the criterion of the ratio of the shortest edge to the longest edge of a triangle. The value *Avg dev*. shows the accuracy of an implicit object approximation and the adaptive ES algorithm is logically the best of tested methods. The criterions of angles and length of edges in triangles are similar for the ES and the MTR algorithms, so the both approaches generate well-shaped triangular meshes.

For visual comparison, the resulting pictures of the Genus 3 object generated in the test are in figures below. Fig. 8a shows the object generated by the adaptive algorithm, so the number of triangles generated is higher in dependence on the surface curvature. In Fig. 8b, some parts of the object are lost because the algorithm just connects nearest parts by large triangles depending of the lowest level of detail. The resulting image generated by the Marching cubes algorithm is shown in Fig. 8c. This algorithm produces badly-shaped triangles but it is fast and also stable for complex implicit surfaces with C^0 continuity, only.

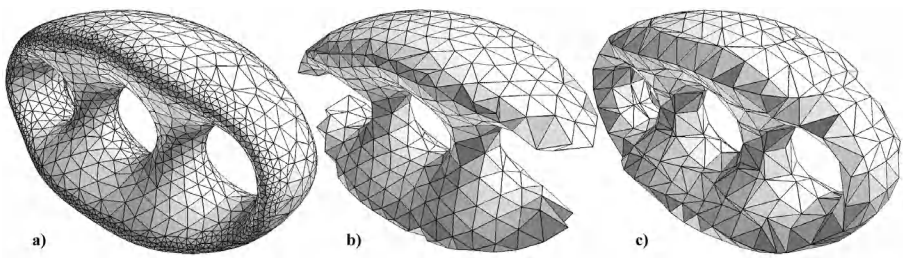


Fig. 8. The Genus 3 object generated by the a) Adaptive Edge spinning algorithm; b) Marching triangles algorithm; c) Marching cubes algorithm.

6 Conclusion

This paper presents the new adaptive approach for polygonization of implicit surfaces. The algorithm marches over the object's surface and computes the accurate coordinates of new points by spinning the edges of already generated triangles. Coordinates of the new points depend on surface curvature estimation. We used the estimation by deviation of angles of adjacent points because it is simple and fast for computation. The similar measurement has been used as curvature estimation in [17] as well. Our experiments also proved its functionality.

The algorithm can polygonize implicit surfaces which comply C^1 continuity, thin objects and some non-complex objects of C^0 continuity (an object should have only sharp edges, no sharp corners or more complex shapes). In future work, we want to modify the current algorithm for more complex implicit functions of the C^0 continuity, only.

Acknowledgement. The authors would like to thank to all who contributed to the development of this new approach, for their comments and suggestions, especially to colleagues MSc. and PhD. students at the University of West Bohemia in Plzen.

References

1. Akkouche, S., Galin, E.: Adaptive Implicit Surface Polygonization using Marching Triangles, Computer Graphic Forum, 20(2): 67–80, 2001.
2. Bloomenthal, J.: Graphics Gems IV, Academic Press, 1994.
3. Bloomenthal, J.: Skeletal Design of Natural Forms, Ph.D. Thesis, 1995.
4. Bloomenthal, J., Bajaj, Ch., Blinn, J., Cani-Gascuel, M-P., Rockwood, A., Wyvill, B., Wyvill, G.: Introduction to implicit surfaces, Morgan Kaufmann, 1997.
5. Čermák, M., Skala, V.: Polygonization by the Edge Spinning, Int. Conf. Algoritmy 2002, Slovakia, ISBN 80-227-1750-9, September 8–13.
6. Čermák, M., Skala, V.: Accelerated Edge Spinning algorithm for Implicit Surfaces, Int. Conf. ICCVG 2002, Zakopane, Poland, ISBN 839176830-9, September 25–29.
7. Hartmann, E.: A Marching Method for the Triangulation of Surfaces, The Visual Computer (14), pp. 95–108, 1998.
8. Hilton, A., Stoddart, A.J., Illingworth, J., Windeatt, T.: Marching Triangles: Range Image Fusion for Complex Object Modelling, Int. Conf. on Image Processing, 1996.
9. "Hyperfun: Language for F-Rep Geometric Modeling", http://cis.k.hosei.ac.jp/~F-rep/
10. Karkanis, T., Stewart, A.J.: Curvature-Dependent Triangulation of Implicit Surfaces, IEEE Computer Graphics and Applications, Volume 21, Issue 2, March 2001.
11. Ohtake, Y., Belyaev, A., Pasko, A.: Dynamic Mesh Optimization for Polygonized Implicit Surfaces with Sharp Features, The Visual Computer, 2002.
12. Pasko, A., Adzhiev, V., Karakov, M., Savchenko,V.: Hybrid system architecture for volume modeling, Computer & Graphics 24 (67–68), 2000.
13. Rvachov, A.M.: Definition of R-functions, http://www.mit.edu/~maratr/rvachev/p1.htm
14. Shapiro, V., Tsukanov, I.: Implicit Functions with Guaranteed Differential Properties, Solid Modeling, Ann Arbor, Michigan, 1999.
15. Taubin, G.: Distance Approximations for Rasterizing Implicit Curves, ACM Transactions on Graphics, January 1994.
16. Triquet, F., Meseure, F., Chaillou, Ch.: Fast Polygonization of Implicit Surfaces, WSCG'2001 Int.Conf., pp. 162, University of West Bohemia in Pilsen, 2001.
17. Velho,L.: Simple and Efficient Polygonization of Implicit Surfaces, Journal of Graphics Tools, 1(2):5–25, 1996.

SOM: A Novel Model for Defining Topological Line-Region Relations

Xiaolin Wang, Yingwei Luo*, and Zhuoqun Xu

Dept. of Computer Science and Technology, Peking University, Beijing, P.R.China, 100871
lyw@pku.edu.cn

Abstract. Topological line-region relations are generally defined by the Ninth-Intersection Model (9IM) or the Dimensionally Extended Ninth-Intersection Model (DE-9IM) in GIS. In the paper, Segment Operator Model (SOM) is introduced to solve the same problem. Let a simple region *R* filter a simple curve *L* and produce a set of curve segments within the exterior, the interior or the borders of *R*. The topological relations between curve segments and *R* are mapped into seven categories: *across*, *stabsin*, *along*, *bowsto*, *sticksto*, *inside* and *disjoint*. SOM is based on counting the curve segments that belong to each of the seven categories. Any topological relations defined in 9IM or DE-9IM can be expressed in SOM. In SOM, *L* is *atomic* to *R* when only a single curve segment is produced, *simplex* to *R* when no more than three curve segments are produced, otherwise, *L* is *complex* to *R*. *L* is *uniform* to *R* when only one kind of curve segments are produced.

1 Introduction

It's importance to identify topological line-region relations in GIS. Nowadays topological line-region relations are generally defined by N-Intersection Model (e.g. Fourth-Intersection Model, 4IM; Ninth-Intersection Model, 9IM), or Dimensionally Extended N-Intersection Model (e.g. DE-9IM). DE-9IM is the most powerful of them [1]. In this paper, Segment Operator Model (SOM) is introduced to identify topological line-region relations instead of those models.

Let the interior, the boundary and the exterior of a simple region *R* to intersect with a simple curve *L*, a set of curve segments will be produced. These curve segments satisfy that the interior of each curve segment intersects with only the exterior, the interior or the borders of *R*. The topological relations between a curve segment and *R* can be divided into seven categories: **across**, **stabsin**, **along**, **bowsto**, **sticksto**, **inside** and **disjoint**.

By counting the curve segments that belong to each of the seven categories, Segment Operator Model (SOM) is built to identify the topological relations of *L* and *R*. Any topological relations of *L* and *R* that can be identified in DE-9IM can also be identified in SOM. In fact, some more complex relations may also be identified in SOM.

* Corresponding author: Yingwei Luo, lyw@pku.edu.cn.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 335–344, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In SOM, *L* is **atomic** to *R* when only a single curve segment is produced, **simplex** to *R* when no more than three curve segments, otherwise, *L* is **complex** to *R*. *L* is **uniform** to *R* when only one kind of curve segments are produced.

In the following sections, related works is introduced in section 2. In section 3, SOM is discussed formally. Section 4 compares SOM with the three intersection models: 4DIM, 9IM, DE-9IM, applied by OGC [1], and then in section 5, we build up a set of named topological relations. The conclusions are in section 6.

2 Related Works

Topological relations of two geometric objects are generally described with N-Intersection Models. N-Intersection Models are based on point set topology theory. Firstly, let's review N-Intersection Models applied in the specifications of OpenGIS. We introduce four operations on geometrical objects: **boundary**, **interior**, **exterior** and **closure**, from [1]. For formalized definitions of them, please refer to [1]. For a region, the boundary of it is all the points on its border line, the interior is all the points inside of the region but excluding its boundary, the exterior is all the points outside of the region, and the closure is the union of its boundary and interior. For a curve, the boundary of it is its two end points, the interior is all the points on the curve but excluding the two end points, the exterior is all the points outside of the curve, and the closure is all the points on the curve including the two end points. If a line is closed, that is to say, if the curve's two end points are the same, the boundary of the line is an empty set. The initial of those operators is used as the abbreviation of them: **b** for **boundary**, **i** for **interior**, **e** for **exterior** and **c** for **closure.**

Three kinds of intersection pattern matrix are used to test the topological relations of two geometric objects in [1]. The first is 4-intersection pattern matrix (4IM). For two objects, A and B the following 4 intersection operation may be done:

$$\begin{pmatrix} A.c \cap B.c & A.c \cap B.e \\ A.e \cap B.c & A.e \cap B.e \end{pmatrix}$$

This matrix of sets may be tested to see if each set is empty or not. This classifies the relationship between A and B into one of 2^4, or 16, classes.

A template may be applied to the intersection matrix to test for a particular spatial relationship between the two objects. The template is a matrix of four extended Boolean Values whose interpretation is given in Table 1. There are 3^4 or 81 possible templates.

Table 1. Meaning of 4-intersection (and Egenhofer intersection) pattern matrix

Symbol	Non Empty?	Meaning
T	TRUE	The intersection at this position of the matrix is non-empty.
F	FALSE	The intersection at this position of the matrix is empty.
N	NULL	This template does not test the intersection at this position of the matrix.

The second is Egenhofer intersection pattern matrix (9IM) that is introduced by Professor Egenhofer of the University of Maine [3]. For two objects, A and B the following 9 intersection operation may be done:

$$\begin{pmatrix} A.b \cap B.b & A.b \cap B.i & A.b \cap B.e \\ A.i \cap B.b & A.i \cap B.i & A.i \cap B.e \\ A.e \cap B.b & A.e \cap B.i & A.e \cap B.e \end{pmatrix}$$

This matrix of sets (called the 9 matrix) may be tested to see if each is empty or not. This classifies the relationship between A and B into one of 2^9, or 512, classes. Actually, not all 512 are geometrically possible, but that is not of consequence to what is to follow.

A template may be applied to the intersection matrix to test for a particular spatial relationship between the two objects. The template is a matrix of nine extended Boolean Values whose interpretation is given in Table 1. There are 3^9 or 19,683 possible templates.

The third is Clementini intersection pattern matrix (DE-9IM) that is similar to the Egenhofer intersection pattern matrix, but a finer distinction is made on the possible values (see Table 2) [4][5].

Table 2. Meaning of Clementini intersection pattern matrix

Symbol	Non Empty?	Meaning
0	TRUE	The intersection at this position of the matrix contains only points.
1	TRUE	The intersection at this position of the matrix contains only points, and curves.
2	TRUE	The intersection at this position of the matrix contains only points, curves, and surfaces.
3	TRUE	The intersection at this position of the matrix contains only points, curves, surfaces and solids.
F	FALSE	The intersection at this position of the matrix is empty.
N	NULL	This template does not test the intersection at this position of the matrix.

To test if two objects are related in agreement with one of the possible 6^9 = 10,077,696 templates, the intersections not associated to NULL are calculated and tested for non-empty and dimension, according to the pattern in the matrix.

Named Topological Relations Set (NTRS) is also used to describe topological relations. In models of NTRS, topological relations are defined in natural language at concept level. Such as in OpenGIS, eight named topological relations based on 9IM are mentioned: *disjoint*, *meet*, *overlaps*, *equal*s, *contains*, *inside*, *covers*, *coveredBy* [2][3]. In CBM (Calculus-based Method), five named topological relations (*touch*, *in*, *cross*, *overlap*, *disjoint*) and three border operators (b, f, t) are used to describe topological relations between simple geometric objects. The capability of CBM is equivalent to DE-9IM. [4] RCC (Region Collection Calculus) describes relations of regions, and is used in spatial reasoning.

Egenhofer introduced metric concepts to measure the degree of *splitting*, *closeness* and *approximate alongness* with respect to the region's area, the line's length, and the region's perimeter. [6] These metric concepts are useful for mapping natural language to computational models of spatial line-region relations. However, the measurement

of *splitting*, *closeness* and *approximate alongness* are not topological unchangeable. [7]

SOM concentrates on relations of curve segments (parts of a line) to the region. It is a new way to describe topological line-region relations.

3 Segmental Operator Model

Let R be a simple region (or surface) on a plane whose boundary is a set of oriented, closed curves that delineates the limits of the region. R's interior is connected which implies that any two direct positions within the surface can be placed on a curve that remains totally inside the region. [1] R is a closure that implies the boundary of R belongs to R. R is also allowed to have holes.

The meanings of notations used in the following sections are listed here:

$\forall$	any	$\cap$	intersects, intersection
$\exists$	exists	$\setminus$	subtracts, subtraction
$\Leftrightarrow$	be equivalent to	$\in$	element of
$\vee$	or	$\subset$	subset of
$\wedge$	and		empty set
$\neg$	not		

Let L be a simple curve, the equation of L is $f(t)=(x(t),y(t)), 0\le t\le 1$, in which, both $x(t)$ and $y(t)$ are continuous. The curve is simple also implies that no interior direct position is involved in a self-intersection of any kind, [1] that is

$$\forall t_1,\forall t_2, 0\le t_1\ne t_2\le 1 \Leftrightarrow f(t_1)\ne f(t_2)\Leftrightarrow x(t_1)\ne x(t_2)\vee y(t_1)\ne y(t_2).$$

L is not a closed curve. The topological relations of a closed curve and R can be identified similarly to SOM. In this paper, we concentrate on a non-closed curve.

Curve segment is the most important concept in SOM. A sub curve of L is called L's curve segment to R, if and only if at least one end of the sub curve sits on the border of R, and the interior of the sub curve belongs to only the exterior, the interior or the border of R; or the sub curve sits properly inside the exterior or the interior. All L's curve segments to R forms a curve segments set S. To define S formally, we introduce the following set S_l, S_e, S_i, and S_b.

Let $L(t_1,t_2)=\{f(t), t_1<t<t_2\}, 0\le t_1<t_2\le 1$ be an opened sub curve of L which do not include the two end points in it. Then let S_l be the set of all the opened sub curves of L. That is

$$S_l=\{L(t_1,t_2)\mid 0\le t_1<t_2\le 1\}.$$

Let S_e be a set of exterior opened sub curves of L to R. The formal definition is

$$S_e=\{l\mid (l\in S_l)\wedge(l\cap R.\mathrm{e}=l)\wedge(\neg\exists l'\in S_l, st.(l'\setminus l\ne\Phi)\wedge(l'\cap R.\mathrm{e}=l'))\}.$$

Let S_i a set of interior opened sub curves of L to R. The formal definition is

$$S_i=\{l\mid (l\in S_l)\wedge(l\cap R.\mathrm{i}=l)\wedge(\neg\exists l'\in S_l, st.(l'\setminus l\ne\Phi)\wedge(l'\cap R.\mathrm{i}=l'))\}.$$

Let S_b a set of bounder opened sub curves of L to R. The formal definition is

$$S_b = \{l \mid (l \in S_l) \wedge (l \cap R.\mathrm{b} = l) \wedge (\neg \exists l' \in S_l, st.(l' \setminus l \neq \Phi) \wedge (l' \cap R.\mathrm{b} = l'))\}.$$

Then S is the set of closures of all opened sub curves those belong to S_e, S_i and S_b. That is

$$S = \{l.\mathrm{c} \mid (l \in S_e) \vee (l \in S_i) \vee (l \in S_b)\}.$$

The sub curve in S is called L's curve segment to R. For each curve segment l, the topological relation between l and R is one of the following: **across**, **stabsin**, **along**, **bowsto**, **sticksto**, **inside** and **disjoint**. Figure 1 illustrates the general mean of these relations. The definitions of them are given bellow:

$\forall l \in S, l$ across $R \Leftrightarrow (l.\mathrm{i} \subset R.\mathrm{i}) \wedge (l.\mathrm{b} \subset R.\mathrm{b})$

$\forall l \in S, l$ stabsin $R \Leftrightarrow (l.\mathrm{i} \subset R.\mathrm{i}) \wedge (l.\mathrm{b} \cap R.\mathrm{b} \neq \Phi) \wedge (l.\mathrm{b} \cap R.\mathrm{i} \neq \Phi)$

$\forall l \in S, l$ along $R \Leftrightarrow (l \subset R.\mathrm{b}) \wedge (l.b \neq \Phi)$

$\forall l \in S, l$ bowsto $R \Leftrightarrow (l.\mathrm{i} \subset R.\mathrm{e}) \wedge (l.\mathrm{b} \subset R.\mathrm{b})$

$\forall l \in S, l$ sticksto $R \Leftrightarrow (l.\mathrm{i} \subset R.\mathrm{e}) \wedge (l.\mathrm{b} \cap R.\mathrm{b} \neq \Phi) \wedge (l.\mathrm{b} \cap R.\mathrm{e} \neq \Phi)$

$\forall l \in S, l$ inside $R \Leftrightarrow (l \subset R.\mathrm{i})$

$\forall l \in S, l$ disjoint $R \Leftrightarrow (l \subset R.\mathrm{e})$

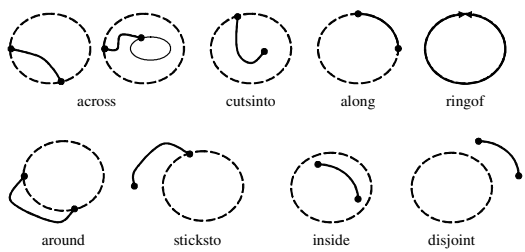


Fig. 1. The relations between a curve segment and R

We prove that those relations are complete, which implies that no other relation exists between l and R. Let's build a decision tree to prove it (see Figure 2). The decision tree shows that if the relation of l and R is not one of the previous seven, l dose not belong to S.

Let **g** be one of the seven relations, we define an operator **g** as a **segment operator** to L and R which works out the total count of l in S that satisfies l **g** R. That is

$$L \text{ g } R = \left|\{l \mid l \text{ g } R\}\right|, \mathrm{g} \in \{\text{stabsin}, \text{sticksto}, \text{arcross}, \text{bowsto}, \text{along}, \text{disjoint}, \text{inside}\}$$

Generally, we have the following constraints:

$$0 \le (L \text{ stabsin } R) + (L \text{ sticksto } R) \le 2$$

$$0 \le (L \text{ inside } R) + (L \text{ disjoint } R) \le 1$$

$$(L \text{ inside } R) + (L \text{ disjoint } R) = 1 \Leftrightarrow (L \text{ across } R) + (L \text{ stabsin } R) + (L \text{ along } R) + (L \text{ bowsto } R) + (L \text{ sticksto } R) = 0$$

For L and R, we define a relation string $\boldsymbol{r}$ as below. And the meanings of $\boldsymbol{r}$ values are listed in Table 3.

$$\boldsymbol{r} = \textbf{inside} \mid \textbf{disjoint} \mid \textbf{ringof} \mid \langle si\rangle\langle st\rangle\langle ac\rangle\langle bw\rangle\langle al\rangle$$

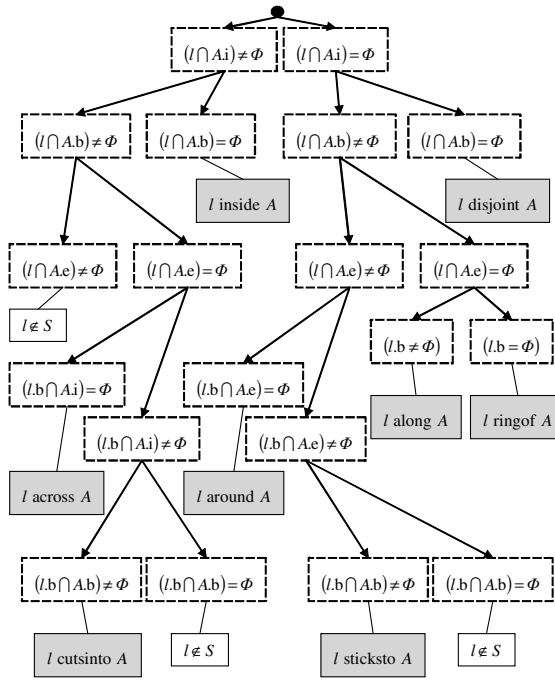


Fig. 2. Completeness of the 7 relations between the curve segment and R

Table 3. Meaning of Segmental curve relation string $\boldsymbol{r}$

Value of $\boldsymbol{r}$	Meaning
inside	(L **inside** R) = 1
disjoint	(L **disjoint** R) = 1
$\langle si\rangle\langle st\rangle\langle ac\rangle\langle bw\rangle\langle al\rangle$	(L **inside** R)+ (L **disjoint** R) = 0
Note: $\langle si\rangle$ = L **stabsin** R, $\langle st\rangle$ = L **sticksto** R, $\langle ac\rangle$ = L **across** R, $\langle bw\rangle$ = L **bowsto** R and $\langle al\rangle$ = L **along** R.	

Figure 3 shows an example how to translate real topological relations to $\boldsymbol{r}$. This also shows that $\boldsymbol{r}$ is more powerful to test the topological relation of L and R than any intersection pattern matrix. For ①, (L **stabsin** R) = 2 and (L **bowsto** R) = 1, all other operators product 0, then $\boldsymbol{r}$ = 20010. For ①, (L **stabsin** R) = 2, (L **across** R) = 1 and (L **bowsto** R) = 2, all other operators product 0, then $\boldsymbol{r}$ = 20120.

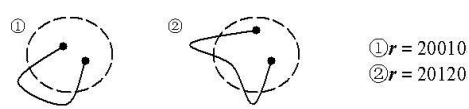


Fig. 3. $\boldsymbol{r}$ values of real topological relations

Theoretically ***r*** values may identify unlimited numbers of topological relations, through not all ***r*** values are meaningful. "+" and "*" may be introduced to the string ***r***. When <si> is "+", it means that (L **stabsin** R) > 0; when <si> is "*", it means that we do NOT care about the value of (L **stabsin** R). The same is applied to <*st*>, <*ac*>, <*bw*> and <*al*>. Then ***r*** values may be used to test a lot of topological relations directly. For example in Figure 3, the two relations belong to ***r*** = 20*+0.

4 Comparing with Intersection Pattern Matrix

For 4-intersection pattern matrix (4IM), only two tests have meanings for L and R. They are shown bellow. The right matrix bellow shows the values that may appear at each place

$$\begin{pmatrix} L.c \cap R.c & L.c \cap R.e \\ \text{T} & \text{T} \end{pmatrix} = \begin{pmatrix} \text{F/T} & \text{F/T} \\ \text{T} & \text{T} \end{pmatrix}$$

The following two equations show that the topological relations of L and R that can be tested with 4-intersection pattern matrix are also can be tested with segment operators.

$$L.\text{c} \cap R.\text{c} = \text{F} \Leftrightarrow (L \text{ disjoint } R) = 1$$

$$L.\text{c} \cap R.\text{e} = \text{F} \Leftrightarrow (L \text{ disjoint } R) + (L \text{ sticksto } R) + (L \text{ bowsto } R) = 0$$

For Egenhofer intersection pattern matrix (9IM), only six tests have meanings for L and R. They are shown bellow. The right matrix bellow shows the values that may appear at each place.

$$\begin{pmatrix} L.\text{b} \cap R.\text{b} & L.\text{b} \cap R.\text{i} & L.\text{b} \cap R.\text{e} \\ L.\text{i} \cap R.\text{b} & L.\text{i} \cap R.\text{i} & L.\text{i} \cap R.\text{e} \\ \text{T} & \text{T} & \text{T} \end{pmatrix} = \begin{pmatrix} \text{F/T} & \text{F/T} & \text{F/T} \\ \text{F/T} & \text{F/T} & \text{F/T} \\ \text{T} & \text{T} & \text{T} \end{pmatrix}$$

The following six equations show that the topological relations of L and R that can be tested with Egenhofer intersection pattern matrix are also can be tested with segment operators in the case of knowing whether R has holes or not.

$$(L.\text{b} \cap R.\text{b} = \text{F}) \Leftrightarrow ((L \text{ disjoint } R) + (L \text{ inside } R) = 1) \vee ((L \text{ stabsin } R) + (L \text{ sticksto } R) = 2)$$

$$(L.\text{b} \cap R.\text{i} = \text{F}) \Leftrightarrow ((L \text{ inside } R) + (L \text{ stabsin } R) = 0)$$

$$(L.\text{b} \cap R.\text{e} = \text{F}) \Leftrightarrow ((L \text{ disjoint } R) + (L \text{ sticksto } R) = 0)$$

$$(L.\text{i} \cap R.\text{b} = \text{F}) \Leftrightarrow ((L \text{ disjoint } R) + (L \text{ inside } R) = 1) \vee ((L \text{ stabsin } R) + (L \text{ sticksto } R) + (L \text{ across } R) + (L \text{ bowsto } R) = 1)$$

$$(L.\text{i} \cap R.\text{i} = \text{F}) \Leftrightarrow ((L \text{ inside } R) = 0) \vee ((L \text{ stabsin } R) + (L \text{ across } R) = 0)$$

$$(L.\text{i} \cap R.\text{e} = \text{F}) \Leftrightarrow ((L \text{ disjoint } R) = 0) \vee ((L \text{ sticksto } R) + (L \text{ bowsto } R) = 0)$$

For Clementini intersection pattern matrix (DE-9IM), only seven tests have meanings for L and R. They are shown bellow. The right matrix bellow shows the values that may appear at each place.

$$\begin{pmatrix} L.\mathrm{b}\cap R.\mathrm{b} & L.\mathrm{b}\cap R.\mathrm{i} & L.\mathrm{b}\cap R.\mathrm{e} \\ L.\mathrm{i}\cap R.\mathrm{b} & L.\mathrm{i}\cap R.\mathrm{i} & L.\mathrm{i}\cap R.\mathrm{e} \\ 1 & 2 & 2 \end{pmatrix} = \begin{pmatrix} \mathrm{F/0} & \mathrm{F/0} & \mathrm{F/0} \\ \mathrm{F/0/1} & \mathrm{F/1} & \mathrm{F/1} \\ 1 & 2 & 2 \end{pmatrix}$$

Comparing with Egenhofer intersection pattern matrix, only two new topological relations are introduced with Clementini intersection pattern matrix. The two can be mapped to segment operators as the following two equations.

$$(L.\mathrm{i}\cap R.\mathrm{b}=0)\Leftrightarrow((L \text{ along } R)=0)\wedge$$
$$((L \text{ stabsin } R)+(L \text{ sticksto } R)+(L \text{ across } R)+(L \text{ bowsto } R)>1)$$
$$(L.\mathrm{i}\cap R.\mathrm{b}=1)\Leftrightarrow((L \text{ along } R)>0)$$

The analysis above shows that, for line-region relations, the capability of SOM is stronger than any of the three-intersection pattern matrix.

SOM are useful to describe some relations directly. For example, in Figure 4, the two distinct relations can be identified by SOM directly (in the first case, the region is whole; while in the second case, the region is split in to tow parts). But for DM-9IM, the Clementini intersection pattern matrixes for the two cases are identical as shown in Figure 4. Some other method should be introduced to distinguish the two cases on the basis of Clementini intersection pattern matrixes.

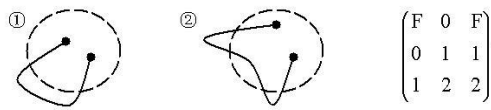


Fig. 4. Two topological relations with the same Clementini intersection pattern matrix

5 Named Topological Relations

In *OpenGIS Simple Features Specification For SQL(Revision 1.1)*, a named topological relation set with eight topological relations are introduced for used in SQL statements for spatial queries, they are *equals*, *disjoint*, *touches*, *within*, *overlaps*, *crosses*, *intersects,* and *contains.* [2] Excluding *overlaps, contains* and *equals* that can not be applied to *L* and *R*, all other topological relations can be expressed in SOM:

$$L \textit{ disjoint } R \Leftrightarrow (L \text{ disjoint } R)=1$$
$$L \textit{ intersects } R \Leftrightarrow (L \text{ disjoint } R)=0$$
$$L \textit{ within } R \Leftrightarrow (L \text{ disjoint } R)+(L \text{ sticksto } R)+(L \text{ bowsto } R)=0$$
$$L \textit{ crosses } R \Leftrightarrow ((L \text{ sticksto } R)+(L \text{ bowsto } R)>0)$$
$$\wedge((L \text{ stabsin } R)+(L \text{ across } R)>0)$$
$$L \textit{ touches } R \Leftrightarrow ((L \text{ sticksto } R)+(L \text{ bowsto } R)=1)$$
$$\wedge((L \text{ stabsin } R)+(L \text{ across } R)+(L \text{ along } R)=0)$$

When S contains only a single curve segment ($|S| = 1$), L is called **atomic** to R. When S has no more than three curve segments ($|S| \leq 3$), L is called **simplex** to R; When there are more than three curve segments, L is called **complex** to R.

We define three topological relations: *som_atomicto*, *som_simplexto* and *som_complexto* for the above cases. Their definitions are:

$$L\ som_atomicto\ R \Leftrightarrow \begin{pmatrix} (L \text{ inside } R)+(L \text{ disjoint } R)+(L \text{ across } R)+ \\ (L \text{ stabsin } R)+(L \text{ along } R)+(L \text{ bowsto } R)+(L \text{ sticksto } R)=1 \end{pmatrix}$$

$$L\ som_simplexto\ R \Leftrightarrow \begin{pmatrix} (L \text{ inside } R)+(L \text{ disjoint } R)+(L \text{ across } R)+ \\ (L \text{ stabsin } R)+(L \text{ along } R)+(L \text{ bowsto } R)+(L \text{ sticksto } R)\leq 3 \end{pmatrix}$$

$$L\ som_complexto\ R \Leftrightarrow \neg(L\ som_simplexto\ R)$$

The three relations are not mutually exclusive to each other, but they are complete. For any L and R, at least one of the three is true.

Seven more topological relations: *som_across*, *som_stabsin*, *som_along*, *som_bowsto*, *som_sticksto*, *som_inside* and *som_disjoint* may be defined based on the seven segment operators. Their definitions are:

$$L\ som_xxx\ R \Leftrightarrow (L \text{ xxx } R) > 0, \text{xxx} \in \begin{Bmatrix} \text{inside, disjoint, across,} \square \\ \text{stabsin, along, bowsto, sticksto} \end{Bmatrix}$$

The seven relations are not mutually exclusive to each other, but they are complete too. For any L and R, at least one of the seven is true. When only one is true, L is called **uniform** to R, and *som_uniformto* is defined for such topological relation.

Table 4. Combinability of named topological relations

	intersects	***crosses***	***within***	***touches***	***disjoint***	***som_disjoint***	***som_inside***	***som_across***	***som_stabsin***	***som_along***	***som_bowsto***	***som_sticksto***	***som_atomicto***	***som_simplexto***	***som_complexto***	***som_uniformto***
intersects	a	m	m	m	-	-	m	m	m	m	m	m	m	m	m	m
crosses	m	a	-	-	-	-	-	m	m	m	m	m	-	m	m	-
within	m	-	a	-	-	-	a	m	m	m	-	-	m	m	m	m
touches	m	-	-	a	-	-	-	-	-	-	m	m	m	m	m	m
disjoint	-	-	-	-	a	a	-	-	-	-	-	-	m	m	-	m
som_disjoint	-	-	-	-	a	a	-	-	-	-	-	-	m	m	-	m
som_inside	m	-	m	-	-	-	a	-	-	-	-	-	m	m	-	m
som_across	m	m	m	-	-	-	-	a	m	m	m	m	m	m	m	m
som_stabsin	m	m	m	-	-	-	-	m	a	m	m	m	m	m	m	m
som_along	m	m	m	-	-	-	-	m	m	a	m	m	m	m	m	m
som_bowsto	m	m	-	m	-	-	-	m	m	m	a	m	m	m	m	m
som_sticksto	m	m	-	m	-	-	-	m	m	m	m	a	m	m	m	m
som_atomicto	m	-	m	m	a	a	a	m	m	m	m	m	a	m	-	m
som_simplexto	m	m	m	m	a	a	a	m	m	m	m	m	a	a	-	m
som_complexto	m	m	m	m	-	-	-	m	m	m	m	m	-	-	a	m
som_uniformto	m	-	m	m	a	a	a	m	m	m	m	m	a	m	m	a

Now with the five topological relations from [2], and the newly defined topological relations, we have totally sixteen relations to describe the topological relations between *L* and *R* (in fact, *som_disjoint* is equivalent to *disjoint*). These relations are not exclusive against each other, the combinations of them will produce many meaningful topological relations that are hardly to express in DE-9IM. Table 4 shows the combinability of those sixteen topological relations. Each column shows that when one of these relations is tenable, which relation *may* combine with it (marked with m), which is always true (marked with a), and which will never happen (marked with -).

6 Conclusions

In this paper, we build up the Segment Operator Model to describe topological relations between a simple curve and a simple region in the same plane. We can conclude that the model is at least equivalent to DE-9IM to describe topological line-region relations. In fact, some more complex topological relations between a curve and a region can be identified in this model. This model might be extended to topological line-volume relations and to topological ring-region or ring-volume relations too, thought they have not been proved in this paper. Whether this model can be extended to describe topological relations between any two geometrical objects is feature work.

Acknowledgement. This work is supported by the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20020001015; the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312000; the National Science Foundation of China under Grant No.60073016 and No.60203002; the National High Technology Development 863 Program under Grant No. 2002AA135330, No. 2002AA134030 and No. 2001AA113151; the Beijing Science Foundation under Grant No.4012007.

References

1. The OpenGIS Abstract Specification—Topic 1: Feature Geometry (ISO 19107 Spatial schema) Version 5. http://www.opengis.org.
2. OpenGIS Simple Features Specification For SQL Revision 1.1. http://www.opengis.org
3. Egenhofer, M.F. and Franzosa: Point Set Topological Spatial Relations, International Journal of Geographical Information Systems, 5(2): 161–174(1991).
4. Clementini E. and Di Felice P.: A Comparison of Methods for Representing Topological Relationships, Information Sciences 80: 1–34(1994).
5. Clementini E. and Di Felice P.: A Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases, Information Sciences 90 (1–4): 121–136(1996).
6. A. R. Shariff, M. J. Egenhofer and D. Mark: Natural-Language Spatial Relations Between Linear and Areal Objects: The Topology and Metric of English-Language Terms, International Journal of Geographic Information Science, 12(3): 215–246(1998).
7. F. Wolter, M. Zakharysacher: Spatial Reasoning in RCC-8 with Boolean Region Terms, In Proceedings of ECAI 2000, P244–250, IOS Press (2000).

On Automatic Global Error Control in Multistep Methods with Polynomial Interpolation of Numerical Solution

Gennady Yu. Kulikov and Sergey K. Shindin

School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa
{gkulikov,sshindin}@cam.wits.ac.za

Abstract. In recent papers [8], [9] the technique for a local and global errors estimation and the local-global step size control were presented to solve ordinary differential equations by means of variable-coefficients multistep methods with the aim to attain automatically any reasonable accuracy set by the user for the numerical solution. Here, we extend those results to the class of multistep formulas with fixed coefficients implemented on nonuniform grids. We give a short theoretical background and numerical examples which clearly show that the local-global step size control works in multistep methods with polynomial interpolation of the numerical solution when the error of interpolation is sufficiently small.

1 Introduction

The problem of an automatic global error control for the numerical solution of ordinary differential equations (ODEs) of the form

$$x'(t) = g\big(t, x(t)\big), \quad t \in [t_0, t_0 + T], \quad x(t_0) = x^0, \tag{1}$$

where $x(t) \in \mathbf{R}^n$ and $g : D \subset \mathbf{R}^{n+1} \to \mathbf{R}^n$ is a sufficiently smooth function, is one of the challenges of modern computational mathematics. ODE (1) is quite usual in applied research and practical engineering (see, for example, [1]–[7]). Therefore any issue in that field possesses a great potential to develop intelligent software packages for mathematical modelling tasks.

In [8] we explored variable-coefficients multistep formulas to change a step size in the course of integration and presented the step size selection mechanism which attains any required accuracy (up to round-off) for the numerical solution. Now we extend those results to fixed-coefficients multistep formulas with Hermite interpolation of the numerical solution. We give the necessary theory and show how the local-global step size control works for the interpolation-type multistep methods in practice.

The paper is organized as follows: Sect. 2 is devoted to the exact definition of multistep formulas with polynomial interpolation of the numerical solution and

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 345–354, 2004.
© Springer-Verlag Berlin Heidelberg 2004

to approximation, stability and convergence properties of such sort of methods. An efficient way to evaluate the local and global errors of the interpolation-type multistep formulas and a short description of the algorithm to select a step size are presented in Sect. 3. A starting procedure for multistep methods of the interpolation type is given in Sect. 4 We test our methods on the restricted three body problem from [6] and discuss the results in the last section of the paper.

2 Multistep Methods of Interpolation Type

Further, we suppose that ODE (1) possesses a unique solution $x(t)$ on the whole interval $[t_0, t_0 + T]$. To solve problem (1) numerically, we apply a stable linear l-step method of order s to obtain the discrete problem

$$\sum_{i=0}^{l} a_i x_{k+1-i} = \tau_k \sum_{i=0}^{l} b_i g(t_{k+1-i}, x_{k+1-i}), \quad k = l-1, l, ..., K-1, \tag{2}$$

where a_i, b_i, $i = 1, 2, \ldots, l$ are real numbers and $a_0 \neq 0$. We consider that the starting values x_k, $k = 0, 1, \ldots, l-1$, are known. Formula (2) implies that the step size τ_k must be fixed. Unfortunately, the latter requirement is too restrictive for many practical problems. Therefore the simplest way to advance the $(k+1)$-st step of a size $\tau_k \neq \tau_{k-1}$ by multistep method (2) is to use an interpolation polynomial for recomputing the numerical solution on the new uniform grid w_{k+1} with the step size τ_k. Thus, when applying the interpolation formula to implement method (2) on a nonuniform grid w_τ with diameter τ we come to the following formal definition:

Definition 1 *A linear multistep method of the form*

$$t_{k+1-i}^{k+1} = t_k - (i-1)\tau_k, \quad x_{k+1-i}^{k+1} = H_{l+1}^{p}(t_{k+1-i}^{k+1}), \quad i = 1, 2, \ldots, l, \tag{3a}$$

$$a_0 x_{k+1} + \sum_{i=1}^{l} a_i x_{k+1-i}^{k+1} = \tau_k b_0 g(t_{k+1}, x_{k+1}) + \tau_k \sum_{i=1}^{l} b_i g(t_{k+1-i}^{k+1}, x_{k+1-i}^{k+1}), \tag{3b}$$

$$t_{k+1}^{k+1} = t_{k+1}, \quad x_{k+1}^{k+1} = x_{k+1}, \quad k = l, l+1, \ldots, K-1, \tag{3c}$$

where $H_{l+1}^{p}(t)$ is the Hermite interpolation formula of degree $p \leq 2s+1$ based on the values x_{k-i}^{k}, $i = 0, 1, \ldots, l$, and $g(t_{k-i}^{k}, x_{k-i}^{k})$, $i = 0, 1, \ldots, p-l-1$, calculated during the k-th step at the nodes t_{k-i}^{k}, $i = 0, 1, \ldots, l$, of the uniform grid w_k with step size τ_{k-1} ($t_k^k = t_k$) is called a linear multistep method with polynomial interpolation of the numerical solution or an interpolation LM method, for short.

Here, we have assumed that the starting values x_{l-i}^{l}, $i = 0, 1, \ldots, l$, are given on a uniform grid w_l with step size τ_{l-1}.

Method (3) can be consider as a combined one. It consists of the procedure to advance a step by underlying multistep formula (2) and of the procedure to change a step size by Hermite interpolation.

To discuss an approximation property of the interpolation LM methods we give

Definition 2 *The function*

$$L\big(t_{k+1},x(t),\tau_k\big)=a_0x(t_{k+1})-\tau_kb_0g(t_{k+1},x(t_{k+1}))\\ +\sum_{i=1}^{l}a_i\tilde{H}^p_{l+1}(t^{k+1}_{k+1-i})-\tau_k\sum_{i=1}^{l}b_ig\big(t^{k+1}_{k+1-i},\tilde{H}^p_{l+1}(t^{k+1}_{k+1-i})\big), \tag{4}$$

$k=l-1,l,\ldots,K-1$, *where the Hermite interpolation formula* $\tilde{H}^p_{l+1}(t)$ *is constructed with the use of the exact solution* $x(t^k_{k-i})$, $i=0,1,\ldots,l$, *and its derivative* $g(t^k_{k-i},x(t^k_{k-i}))$, $i=0,1,\ldots,p-l-1$, *calculated on the uniform grid* w_k *with step size* τ_{k-1} *is called the defect of method* (3).

Then the theoretical result for the approximation is quite evident:

$$\left.\begin{array}{l}1)\ \text{Underlying method (2) is of order } s.\\ 2)\ \text{Hermite interpolation is of degree } p.\\ 3)\ \text{Step size ratios } \tau_k/\tau_{k-1} \text{ are bounded.}\end{array}\right\}\Rightarrow L\big(t_{k+1},x(t),\tau_k\big)=O(\tau^{\min\{s,p\}+1})$$

where $\tau=\max_{0\le k\le K-1}\{\tau_k\}$. In addition, if $p>s$ then we obtain

$$L\big(t_{k+1},x(t),\tau_k\big)\cong\frac{(-1)^{s+1}}{(s+1)!}\sum_{i=1}^{l}\Big(a_ii^{s+1}+(s+1)b_ii^s\Big)x^{(s+1)}(t_{k+1})\tau_k^{s+1}, \tag{5}$$

$k=0,1,\ldots,K-1$, with the error of $O(\tau^{s+2})$.

Stability analysis of the interpolation LM methods is more complicated. First of all we have to rewrite combined method (3) in the form of an one-step higher dimension one. For this purpose, we introduce matrices of dimension $n(l+1)\times n(l+1)$:

$$\bar{U}_1=\begin{pmatrix}-\dfrac{a_1}{a_0} & -\dfrac{a_2}{a_0} & \cdots & -\dfrac{a_l}{a_0} & 0\\ 1 & 0 & \cdots & 0 & 0\\ 0 & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & 1 & 0\end{pmatrix}\quad\text{for the underlying multistep formula,}$$

$$H_1(k)=\begin{pmatrix}h^{(1)}_{00} & h^{(1)}_{01} & \cdots & h^{(1)}_{0l}\\ h^{(1)}_{10} & h^{(1)}_{11} & \cdots & h^{(1)}_{1l}\\ \vdots & \vdots & \ddots & \vdots\\ h^{(1)}_{l0} & h^{(1)}_{l1} & \cdots & h^{(1)}_{ll}\end{pmatrix}\quad\text{for the Hermite interpolation polynomial.}$$

More precisely, the entries of the matrix $H_1(k)$ are defined by formulas

$$h^{(1)}_{ij}=\left(1-(t^{k+1}_{k-i}-t^k_{k-j})\frac{H'_{j,l+1}(t^k_{k-j})}{H_{j,l+1}(t^k_{k-j})}\right)\frac{H_{j,l+1}(t^{k+1}_{k-i})}{H_{j,l+1}(t^k_{k-j})},\quad j=0,\ldots,p-l-1,$$

$$h_{ij}^{(1)} = \frac{H_{j,l+1}(t_{k-i}^{k+1})}{H_{j,l+1}(t_{k-j}^{k})}, \quad j = p-l, \ldots, l,$$

where

$$H_{i,l+1}(t) = \prod_{m=0, m\neq i}^{p-l-1} (t - t_{k-m}^{k})^2 \prod_{m=p-l, m\neq i}^{l} (t - t_{k-m}^{k}), \quad i = 0, 1, \ldots, l.$$

This follows from the form of a Hermite interpolation polynomial in [3, p. 172].

Thus, having applied the one-step method, which is equivalent to (3), to the trivial ODE

$$x'(t) = 0, \quad x(t_0) = x^0$$

on a grid $w_\tau \in \mathbf{W}_{\omega_1,\omega_2}^{\infty}(t_0, t_0+T)$; i.e., on a grid with the step size ratios restricted as shown by the formula

$$0 < \omega_1 \leq \tau_k/\tau_{k-1} \leq \omega_2 < \infty, \quad k = 1, 2, \ldots, K-1, \tag{6}$$

we come to the formal definition:

Definition 3 *Method* (3) *is stable on the set of grids* $\mathbf{W}_{\omega_1,\omega_2}^{\infty}(t_0, t_0+T)$ *if on any grid* w_τ *from this set it is valid that*

$$\left\| \prod_{j=0}^{m} \bar{U}_1 H_1(k-j) \right\| \leq R, \ m = 0, 1, \ldots, k+l-1, \ k = l, l+1, \ldots, K-1, \tag{7}$$

where R *is a constant.*

The symbol "∞" means that the ratio of the maximum step size to the minimum one may be unlimited on grids belonging to the set $\mathbf{W}_{\omega_1,\omega_2}^{\infty}(t_0, t_0+T)$ when the diameter $\tau \to 0$.

Definition 3 says that the stability of the interpolation LM methods depends on the constants ω_1 and ω_2. It is interesting to determine magnitudes of these constants for Backward Differentiation Formulas (BDFs) and for implicit Adams methods. We will see that the stability of the underlying method does (2) not guarantee the stability of method (3) on any nonuniform grid.

For example, let us consider the stability problem for the BDF of order 2 with the Hermite interpolation polynomial of degree 3. We remember that any implicit BDF is stable if its order does not exceed 6 (see Theorem 3.4 in [6]). Following now Definition 3 we find the matrices

$$\bar{U}_1 = \begin{pmatrix} \frac{4}{3} & -\frac{1}{3} & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad H_1(k) = \begin{pmatrix} 1 & 0 & 0 \\ 1 - \frac{\theta_k^2}{4}(7-3\theta_k) & \theta_k^2(2-\theta_k) & -\frac{\theta_k^2}{4}(1-\theta_k) \\ 1 - \theta_k^2(7-6\theta_k) & 8\theta_k^2(1-\theta_k) & -\theta_k^2(1-2\theta_k) \end{pmatrix}$$

where $\theta_k \stackrel{\text{def}}{=} \tau_k/\tau_{k-1}$. It is easy to compute that eigenvalues of the companion matrix $\bar{U}_1 H_1(k)$ are: 0, 1 and $\theta_k^2/3$. Thus, the interpolation LM method based

on the BDF of order 2 and on the Hermite interpolation polynomial of degree 3 will be stable if the grid w_τ satisfies the condition

$$0 < \theta_k = \tau_k/\tau_{k-1} < \sqrt{3}.$$

The last inequality says that there exists no lower bound for step size ratios when the interpolation two-step BDF is implemented. This conclusion remains correct for any stable BDF of the order from 2 up to 6, but the upper bounds of step size ratios for the interpolation methods based on stable BDFs exist and they are presented in Table 1. The line in Table 1 implies that there is no interpolation l-step method when $p = l+3$ and $l = 2$ (see Definition 1). Note that the stability requirements ω_1 and ω_2 are more severe in the case of variable-coefficients BDFs than in the case of fixed-coefficients BDFs with the Hermite interpolation of the numerical solution when the order of underlying method is 3 or higher (compare Table 1 with Table 5.1 in [6]). This is a good result for practice.

Table 1. Upper bounds of step size ratios ω_2 for stable interpolation l-step BDFs

l	2	3	4	5	6
$p = l+1$	1.720	1.406	1.241	1.130	1.052
$p = l+2$	1.904	1.525	1.161	1.045	1.012
$p = l+3$	—	1.341	1.090	1.026	1.009

We remark that implicit Adams methods of the interpolation type are stable on any nonuniform grid w_τ. The latter follows immediately from the definitions of interpolation LM methods and of implicit Adams methods (see, for instance, [6]).

We complete Sect. 2 with the convergence result for method (3).

Theorem 1 *Let the right-hand part of problem* (1) *be sufficiently differentiable in a neighborhood of the solution* $x(t)$ *on the interval* $[t_0, t_0+T]$ *and interpolation LM method* (3) (*based on an underlying* l*-step formula of order* s *and the Hermite polynomial* $H^p_{l+1}(t)$ *of degree* p) *be stable on a set of grids* $\mathbf{W}^\infty_{\omega_1,\omega_2}(t_0, t_0+T)$. *Suppose that the starting values* x_i, $i = 0, 1, \ldots, l-1$, *are given with accuracy* $O(\tau^{\min\{s,p\}})$. *Then method* (3) *converges with order* $\min\{s, p\}$ *to the exact solution of ODE* (1) *on grids* w_τ, *with a sufficiently small diameter* τ, *belonging to the set* $\mathbf{W}^\infty_{\omega_1,\omega_2}(t_0, t_0+T)$; *i.e.,*

$$\|x(t_i) - x_i\| \le C\tau^{\min\{s,p\}}, \quad i = 0, 1, \ldots, K.$$

where C *is a constant.*

The proof of Theorem 1 and full particulars of the theory of interpolation LM methods will appear in [12].

Table 2. Coefficients c_j for implicit l-step BDFs

l	c_0	c_1	c_2	c_3	c_4	c_5	c_6
2	$\frac{-1}{3}$	$\frac{2}{3}$	$\frac{-1}{3}$				
3	$\frac{-1}{4}$	$\frac{3}{4}$	$\frac{-3}{4}$	$\frac{1}{4}$			
4	$\frac{-1}{5}$	$\frac{4}{5}$	$\frac{-6}{5}$	$\frac{4}{5}$	$\frac{-1}{5}$		
5	$\frac{-1}{6}$	$\frac{5}{6}$	$\frac{-10}{6}$	$\frac{10}{6}$	$\frac{-5}{6}$	$\frac{1}{6}$	
6	$\frac{-1}{7}$	$\frac{6}{7}$	$\frac{-15}{7}$	$\frac{20}{7}$	$\frac{-15}{7}$	$\frac{6}{7}$	$\frac{-1}{7}$

Table 3. Coefficients c_j and d_j for implicit l-step Adams methods

l	c_0	c_1	c_2	c_3	c_4	c_5	c_6	d_0
1	$\frac{1}{6}$	$\frac{-1}{6}$						$\frac{-1}{6}$
2	$\frac{3}{16}$	$\frac{-4}{16}$	$\frac{1}{16}$					$\frac{-2}{16}$
3	$\frac{209}{1080}$	$\frac{-342}{1080}$	$\frac{171}{1080}$	$\frac{-38}{1080}$				$\frac{-114}{1080}$
4	$\frac{25}{128}$	$\frac{-48}{128}$	$\frac{36}{128}$	$\frac{-16}{128}$	$\frac{3}{128}$			$\frac{-12}{128}$
5	$\frac{118231}{604800}$	$\frac{-258900}{604800}$	$\frac{258900}{604800}$	$\frac{-172600}{604800}$	$\frac{64725}{604800}$	$\frac{-10356}{604800}$		$\frac{-51780}{604800}$
6	$\frac{8085}{41472}$	$\frac{-19800}{41472}$	$\frac{24750}{41472}$	$\frac{-22000}{41472}$	$\frac{12375}{41472}$	$\frac{-3960}{41472}$	$\frac{550}{41472}$	$\frac{-3300}{41472}$

3 Local and Global Errors Estimation

We start with some restriction on the set of admissible grids $\mathbf{W}^{\infty}_{\omega_1,\omega_2}(t_0, t_0+T)$. Below, we consider only grids $w_\tau \in \mathbf{W}^{\Omega}_{\omega_1,\omega_2}(t_0, t_0+T) \subset \mathbf{W}^{\infty}_{\omega_1,\omega_2}(t_0, t_0+T)$; i.e.,

$$\tau/\tau_k \le \Omega < \infty, \quad k = 0, 1, \ldots, K-1. \tag{8}$$

The last requirement is quite practical because it implies that the ratio of the maximum step size to the minimum one is bounded with the constant Ω. On the other hand, any code solving real life problems must be provided with limits for the maximum step size and for the minimum one, that is equivalent to (8), because of an asymptotic form of the theory of interpolation LM methods and round-off errors.

If we now assume that $p > s \geq 2$ and the starting values x_{l-i}^l, $i = 0, 1, \ldots, l$, are given on a uniform grid of step size τ_{l-1} with an accuracy of $O(\tau_{l-1}^{s+2})$. In the next section, we will discuss how to do that in practice. Then the following formulas compute the principal terms of the local and global errors of stable interpolation LM method (3):

$$\begin{aligned}\Delta\tilde{x}_{k+1} \cong &\Big(a_0 I_n - \tau_k b_0 \partial_x g(t_{k+1}, \tilde{x}_{k+1})\Big)^{-1} \Bigg(\tau_k \sum_{j=0}^{l} c_j g(t_{k+1-j}^{k+1}, \tilde{x}_{k+1-j}^{k+1}) \\ &+\tau_k^2 \sum_{j=0}^{s-l-1} d_j \Big(\partial_t g(t_{k+1-j}^{k+1}, \tilde{x}_{k+1-j}^{k+1}) + \partial_x g(t_{k+1-j}^{k+1}, \tilde{x}_{k+1-j}^{k+1}) \\ &\times g(t_{k+1-j}^{k+1}, \tilde{x}_{k+1-j}^{k+1})\Big)\Bigg),\end{aligned} \tag{9}$$

$$\begin{aligned}\Delta x_{k+1} \cong &\Big(a_0 I_n - \tau_k b_0 \partial_x g(t_{k+1}, \tilde{x}_{k+1})\Big)^{-1} \sum_{i=1}^{l} \Big(\tau_k b_i \partial_x g(t_{k+1-i}^{k+1}, x_{k+1-i}^{k+1}) \\ &-a_i I_n\Big)\Big(\tilde{H}_{l+1}^{s+1}(t_{k+1-i}^{k+1}) - H_{l+1}^{s+1}(t_{k+1-i}^{k+1})\Big) + \Delta\tilde{x}_{k+1},\end{aligned} \tag{10}$$

where c_j, d_j, $j = 0, 1, \ldots, l$, are constants (see [11]), and the Hermite interpolation formula $\bar{H}_{l+1}^{s+1}(t)$ is based on the corrected values $\tilde{x}_{k-i}^k$, $i = 0, 1, \ldots, l$, obtained by a method of order $s+1$ and on $g(t_{k-i}^k, \tilde{x}_{k-i}^k)$, $i = 0, 1, \ldots, s-l$, as well. In formula (9), $\tilde{x}_{k+1-i}^{k+1} = \bar{H}_{l+1}^{s+1}(t_{k+1-i}^{k+1})$. The constants c_j, d_j for implicit BDFs and implicit Adams methods are presented in Tables 2 and 3, respectively. A fuller version of the local and global errors estimation for the interpolation LM methods with necessary details and proofs will appear in [12].

With a practical standpoint, the most important and difficult question is an automatic control of the error arising in the real numerical integration. To treat this problem for interpolation LM methods of the form (3), we apply the local-global step size selection presented in [8]. For short, if we fix ϵ_l and ϵ_g as tolerances for the local and global errors, respectively, and choose the maximum step size τ then that algorithm can be given as follows:

Step 1. $k := l - 1$, $M := 0$; {we set $\tau < 1$ and suppose that $\tau_{l-1} \leq T/l$}
Step 2. If $t_k < t_0 + T$,
then go to *Step 3*,
else go to *Step 13*;
Step 3. $t_{k+1} := t_k + \tau_k$, compute $\tilde{x}_{k+1}$, $\Delta\tilde{x}_{k+1}$;
Step 4. $\tau_k^* := \tau_k \left(\epsilon_l / \|\Delta\tilde{x}_{k+1}\|\right)^{1/(s+1)}$;
Step 5. If $\|\Delta\tilde{x}_{k+1}\| > \epsilon_l$,
then $\tau_k := \tau_k^*$, go to *Step 3*;
Step 6. Compute x_{k+1}, Δx_{k+1};
Step 7. $\tau_k^{**} := \tau_k \Big((\epsilon_g - \|\Delta\tilde{x}_{k+1}\|)/\|\Delta x_{k+1}\|\Big)^{1/s}$;
Step 8. If $\|\Delta x_{k+1}\| \leq \epsilon_g$,
then go to *Step 12*;

Step 9. $\tau_k := \tau_k^{**}$, $M := M + 1$;
Step 10. If $M < 2$,
then go to *Step 3*;
Step 11. $\tau := \tau\,(\epsilon_g/\|\Delta x_{k+1}\|)^{1/s}$, go to *Step 1*;
Step 12. $\tau_{k+1} := \min\{\tau, \tau_k^*, \tau_k^{**}, t_0 + T - t_{k+1}\}$,
$k := k + 1$, $M := 0$, go to *Step 2*;
Step 13. Stop.

Here, we suggest additionally to use safety factors in *Steps 4, 7, 11* of the algorithm and provide condition (6) for all step size changes in order to be in the set of admissible grids $\mathbf{W}^{\Omega}_{\omega_1,\omega_2}(0,T)$. We also introduce a lower step size restriction $\tau_{\min}$ because of round-off errors and consider that ODE (1) cannot be solved numerically with the tolerances ϵ_l, ϵ_g if the local-global step size control has required a step size smaller than $\tau_{\min}$.

4 Starting Procedure

To find uniformly placed starting values with the accuracy of $O(\tau_{l-1}^{s+2})$, we apply the following algorithm:

1. We set an initial step size τ_{l-1}.
2. We take the harmonic sequence $1, 2, 3, 4, 5, 6, 7, 8, \ldots$ and use Extrapolated Mid-Point Rule (EMPR) with the extrapolation number $q_1 = [(s+1)/2]$ (the square brackets mean the integer part of the number) to obtain vectors $T^k_{q_1,q_1}$ at the time points $t_k = t_0 + k\,\tau_{l-1}$, $k = 0, 1, \ldots, l-1$.
3. We apply EMPR once again, but with the greater extrapolation number $q_2 = [(s+1)/2] + 1$ to derive an additional vectors $T^k_{q_2,q_2}$ at the same time points t_k. Thus, it is easy to see that
$$x(t_k) - T^k_{q_1,q_1} = T^k_{q_2,q_2} - T^k_{q_1,q_1} + O(\tau_{l-1}^{s+4}) \quad \text{for even} \quad s, \tag{11a}$$
$$x(t_k) - T^k_{q_1,q_1} = T^k_{q_2,q_2} - T^k_{q_1,q_1} + O(\tau_{l-1}^{s+5}) \quad \text{for odd} \quad s, \tag{11b}$$
$k = 0, 1, \ldots, l-1$, where $T^k_{q_2,q_2} - T^k_{q_1,q_1}$ is an estimate of the principal term of the error of EMPR (when the extrapolation number is $q_1 = [(s+1)/2]$) with the accuracy of $O(\tau_{l-1}^{s+4})$ or of $O(\tau_{l-1}^{s+5})$, respectively.
4. If the condition
$$\|T^k_{q_2,q_2} - T^k_{q_1,q_1}\| \le \epsilon_g$$
holds for any $k = 0, 1, \ldots, l-1$, then we consider that the starting values $T^k_{q_2,q_2}$ have been computed with the zero errors and stop the algorithm.
5. In the opposite case, we calculate the new step size
$$\tau^*_{l-1} := \tau_{l-1}\left(\epsilon_g \Big/ \max_{k=0,1,\ldots,l-1} \|T^k_{q_2,q_2} - T^k_{q_1,q_1}\|\right)^{1/(s+2)} \quad \text{for even } s, \tag{12a}$$
$$\tau^*_{l-1} := \tau_{l-1}\left(\epsilon_g \Big/ \max_{k=0,1,\ldots,l-1} \|T^k_{q_2,q_2} - T^k_{q_1,q_1}\|\right)^{1/(s+3)} \quad \text{for odd } s \tag{12b}$$
and repeat the whole starting procedure once again.

To the end, we recommend to use a safety factor in formulas (12) and refer to [10] for particulars of the theory of implicit extrapolation and for a sufficient number of iteration steps in EMPR to preserve the asymptotics of formulas (11).

5 Numerical Experiments

In this section we give a number of numerical examples confirming the power of the local-global step size control in interpolation LM methods (3). As a test problem, we take the restricted three body problem from [6] which possesses the periodic solution-path:

$$x_1''(t) = x_1(t) + 2x_2'(t) - \mu_1 \frac{x_1(t)+\mu_2}{y_1(t)} - \mu_2 \frac{x_1(t)-\mu_1}{y_2(t)}, \tag{13a}$$

$$x_2''(t) = x_2(t) - 2x_1'(t) - \mu_1 \frac{x_2(t)}{y_1(t)} - \mu_2 \frac{x_2(t)}{y_2(t)}, \tag{13b}$$

$$y_1(t) = \Big((x_1(t)+\mu_2)^2 + x_2(t)^2\Big)^{3/2}, \quad y_2(t) = \Big((x_1(t)-\mu_1)^2 + x_2(t)^2\Big)^{3/2}, \tag{13c}$$

where $t \in [0, T]$, $T = 17.065216560157962558891$, $\mu_1 = 1 - \mu_2$ and $\mu_2 = 0.012277471$. The initial values of problem (13) are: $x_1(0) = 0.994$, $x_1'(0) = 0$, $x_2(0) = 0$, $x_2'(0) = -2.001585106379082522 40$. Thus, we are capable to observe the work of our methods in practice.

Table 4. Global errors obtained for fixed-coefficients implicit l-step BDFs with Hermite interpolation $H_{l+1}^{l+1}(t)$ and with the local-global step size control

l	required accuracy				
	$\epsilon_g = 10^{-02}$	$\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$
2	$5.382 \cdot 10^{-05}$	$7.224 \cdot 10^{-07}$	—	—	—
3	$4.158 \cdot 10^{-04}$	$1.630 \cdot 10^{-05}$	$1.020 \cdot 10^{-06}$	$4.216 \cdot 10^{-08}$	$4.095 \cdot 10^{-09}$
4	$8.703 \cdot 10^{-05}$	$2.756 \cdot 10^{-06}$	$6.849 \cdot 10^{-08}$	$3.838 \cdot 10^{-09}$	$2.336 \cdot 10^{-09}$
5	$1.709 \cdot 10^{-03}$	$2.104 \cdot 10^{-04}$	$1.735 \cdot 10^{-05}$	$1.949 \cdot 10^{-06}$	$1.941 \cdot 10^{-07}$
6	$1.121 \cdot 10^{-03}$	$3.499 \cdot 10^{-05}$	$1.988 \cdot 10^{-06}$	$1.096 \cdot 10^{-07}$	$4.784 \cdot 10^{-07}$

Table 5. Global errors obtained for fixed-coefficients implicit l-step Adams methods with Hermite interpolation $H_{l+1}^{l+2}(t)$ and with the local-global step size control

l	required accuracy				
	$\epsilon_g = 10^{-02}$	$\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$
1	$2.108 \cdot 10^{-05}$	$2.421 \cdot 10^{-07}$	—	—	—
2	$3.152 \cdot 10^{-04}$	$1.079 \cdot 10^{-05}$	$5.032 \cdot 10^{-07}$	$2.014 \cdot 10^{-08}$	$1.137 \cdot 10^{-09}$
3	$5.407 \cdot 10^{-05}$	$2.702 \cdot 10^{-06}$	$1.693 \cdot 10^{-07}$	$8.702 \cdot 10^{-09}$	$3.182 \cdot 10^{-10}$
4	$1.261 \cdot 10^{-04}$	$1.937 \cdot 10^{-05}$	$2.681 \cdot 10^{-06}$	$2.464 \cdot 10^{-07}$	$2.681 \cdot 10^{-08}$
5	$4.106 \cdot 10^{-03}$	$1.745 \cdot 10^{-05}$	$8.490 \cdot 10^{-07}$	$5.599 \cdot 10^{-08}$	$3.372 \cdot 10^{-09}$
6	$1.760 \cdot 10^{-04}$	$7.598 \cdot 10^{-06}$	$8.821 \cdot 10^{-08}$	$2.801 \cdot 10^{-08}$	$4.001 \cdot 10^{-09}$

Now we apply both implicit BDFs and Adams methods of the form (3) to compute the numerical solution of problem (13). Here, we take $\epsilon_l = \epsilon_g^{1+1/s}$. We determine the real errors appeared in the integrations and compare them with the set accuracy. Lines in Tables 4, 5 mean that the second order methods are not able to calculate the numerical solution with the necessary accuracy when $\epsilon_g \leq 10^{-04}$ because the required step size in this situation is too small.

Tables 4, 5 display that all our interpolation methods have achieved the goal; i.e., they have computed the numerical solution of the restricted three body problem with the set accuracy ϵ_g. Thus, we conclude that the local-global step size selection, in fact, allows the global error of method (3) to be controlled in the course of numerical integration. This is a good result to implement it in practice.

References

1. Arushanyan, O.B., Zaletkin, S.F.: Numerical solution of ordinary differential equations using FORTRAN. (*in Russian*) Mosk. Gos. Univ., Moscow, 1990
2. Bakhvalov, N.S.: Numerical methods. (*in Russian*) Nauka, Moscow, 1975
3. Berezin, I.S., Zhidkov, N.P.: Methods of computations. V. 1. (*in Russian*) Gos. izd-vo fiz.-mat. lit-ry, Moscow, 1962
4. Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley and Son, Chichester, 2003
5. Gear, C.W.: Numerical initial value problems in ordinary differential equations. Prentice-Hall, 1971
6. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I: Nonstiff problems. Springer-Verlag, Berlin, 1987
7. Hairer, E., Wanner, G.: Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996
8. Kulikov, G.Yu., Shindin, S.K.: A technique for controlling the global error in multistep methods. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. **40** (2000) No. 9, 1308–1329; *translation in* Comput. Math. Math. Phys. **40** (2000) No. 9, 1255–1275
9. Kulikov, G.Yu., Shindin, S.K.: On multistep extrapolation methods for ordinary differential equations. (*in Russian*) Dokl. Akad. Nauk, **372** (2000) No. 3, 301–304; *translation in* Doklady Mathematics, **61** (2000) No. 3, 357–360
10. Kulikov, G.Yu.: On implicit extrapolation methods for ordinary differential equations. Russian J. Numer. Anal. Math. Modelling. **17** (2002) No. 1, 41–69
11. Kulikov, G.Yu., Shindin, S.K.: On effective computation of asymptotically correct estimates of the local and global errors for multistep methods with fixed coefficients. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. (to appear); *translation in* Comput. Math. Math. Phys. (to appear)
12. Kulikov, G.Yu., Shindin, S.K.: On interpolation type multistep methods with automatic global error control. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. (to appear); *translation in* Comput. Math. Math. Phys. (to appear)

Approximation Algorithms for k-Source Bottleneck Routing Cost Spanning Tree Problems
(Extended Abstract)

Yen Hung Chen[1], Bang Ye Wu[2], and Chuan Yi Tang[1]

[1] Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.
{dr884336,cytang}@cs.nthu.edu.tw
[2] Department of Computer Science and Information Engineering, Shu-Te University, YenChau Kaoshiung 824, Taiwan, R.O.C.
bangye@mail.stu.edu.tw

Abstract. In this paper, we investigate two spanning tree problems of graphs with k given sources. Let $G = (V, E, w)$ be an undirected graph with nonnegative edge lengths and $S \subset V$ a set of k specified sources. The first problem is the *k-source bottleneck vertex routing cost spanning tree* (k-BVRT) problem, in which we want to find a spanning tree T such that the maximum total distance from any vertex to all sources is minimized, i.e., we want to minimize $\max_{v \in V} \left\{\sum_{s \in S} d_T(s, v)\right\}$, in which $d_T(s, v)$ is the length of the path between s and v on T. The other problem is the *k-source bottleneck source routing cost spanning tree* (k-BSRT) problem, in which the objective function is the maximum total distance from any source to all vertices, i.e., $\max_{s \in S} \left\{\sum_{v \in V} d_T(s, v)\right\}$. In this paper, we present a polynomial time approximation scheme (PTAS) for the 2-BVRT problem. For the 2-BSRT problem, we first give $(2 + \varepsilon)$-approximation algorithm for any $\varepsilon > 0$, and then present a PTAS for the case that the input graphs are restricted to metric graphs. Finally we show that there is a simple 3-approximation algorithm for both the two problems with arbitrary k.

Keywords: Combinatorial optimization problem, spanning tree, approximation algorithm, polynomial time approximation scheme

1 Introduction

Finding spanning trees of a given graph is an important problem in network design. Depending on the applications, problems are defined by different objectives. For example, a minimum spanning tree is the spanning tree of minimum total edge weight, and the objective function of the *minimum routing cost spanning tree* (MRCT) [9] is the total distance summed over all pairs of vertices.

Motivated by the applications of multicasting and broadcasting, several multi-source spanning tree problems have been studied [2,4,10]. In such problems, we are given an undirected graph $G = (V, E, w)$ with nonnegative edge

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 355–366, 2004.
© Springer-Verlag Berlin Heidelberg 2004

length function w and $S \subset V$ a set of k specified sources, and asked for the spanning tree minimizing some distance related cost metric. The k-source MRCT (k-MRCT), also called as the k-source shortest path spanning tree, is the spanning tree T minimizing $\sum_{s\in S}\{\sum_{v\in V} d_T(s,v)\}$, where $d_T(s,v)$ is the length of the path between s and v in T. If there is only one source, the problem can be solved by finding the *shortest path tree*, a spanning tree in which the path between the source and each vertex is a shortest path on the given graph. The shortest path tree problem has been well studied and efficient algorithms were developed (see [3] or other text books for algorithms). The most efficient algorithms run in $O(|E| + |V| \log |V|)$ time on graphs with nonnegative edge lengths. And Thorup gave an $O(|E|)$ time algorithm for graphs with positive integer lengths [7].

In the case of more than one sources, the k-MRCT problem is a generalization of the MRCT problem (also called the *shortest total path length spanning tree* problem, [ND3] in [5]), in which all vertices are sources. The MRCT problem is NP-hard [5,6] and admits a *polynomial time approximation scheme* (PTAS) [9]. The k-MRCT problem for general k is obviously NP-hard since it includes the MRCT problem as a special case. However, the NP-hardness of the MRCT does not imply the complexity of the k-MRCT problem for a fixed constant k. The NP-hardness of the 2-MRCT problem was shown independently in [4] and [10]. In the latter paper, a PTAS was also proposed. Recently, the k-MRCT problem for any constant k was also shown to be NP-hard [11] and a 2-approximation algorithm can be found in the previous work of the author [8].

While the k-MRCT problem is defined by a min-sum objective function, Connamacher and Proskurowski posted two variants of k-MRCT problem with bottleneck objective functions. They are the *k-source maximum vertex shortest paths spanning tree* (k-MVST) problem and the *k-source maximum source shortest paths spanning tree* (k-MSST) problem [2]. The objective function of the k-MVST and k-MSST problems are $\max_{v\in V}\{\sum_{s\in S} d_T(s,v)\}$ and $\max_{s\in S}\{\sum_{v\in V} d_T(s,v)\}$ respectively. Both the problems are shown to be NP-Complete in the strong sense even for $k = 2$ [2]. To exhibit the objective and associate with the previous work, we shall use the name *k-source bottleneck vertex routing cost spanning tree* (k-BVRT) for k-MVST and *k-source bottleneck source routing cost spanning tree* (k-BSRT) for k-MSST.

In this paper, we focus on the approximation algorithms for the two problems. For the 2-BVRT problem, we present a PTAS. For the 2-BSRT problem, we show a $(2+\varepsilon)$-approximation algorithm for the general graphs, and a PTAS for metric graphs. A metric graph is a complete graph in which the edge lengths obeying the triangle inequality. Finally, for any arbitrary $k > 2$, we give simple 3-approximation algorithms for both the two problems.

2 Preliminaries

In this paper, a graph is simple, connected and undirected. By $G = (V, E, w)$, we denote a graph G with vertex set V, edge set E and edge length function w. The edge length function is assumed to be nonnegative. For any graph G, $V(G)$

denotes its vertex set and $E(G)$ denotes its edge set. For a subgraph H of G, we define $w(H) = w(E(H)) = \sum_{e \in E(H)} w(e)$. We shall also use n to denote $|V(G)|$. The following notations are used in this paper. For $u, v \in V$, $SP_G(u, v)$ denotes a shortest path between u and v on G. The shortest path length is denoted by $d_G(u, v) = w(SP_G(u, v))$. Let H be a subgraph of G. For a vertex $v \in V(G)$, we let $d_G(v, H)$ denote the shortest distance from v to H, i.e., $d_G(v, H) = \min_{u \in V(H)}\{d_G(v, u)\}$. The definition also includes the case that H is a vertex set but no edge. Let T be a spanning tree of G. For $v \in V$ and a vertex set $R \subset V$, we use $D_T(v, R)$ to denote the total distance from vertex v to R, i.e., $D_T(v, R) = \sum_{r \in R} d_T(v, r)$.

Definition 1. *Let T be a tree and $S \subset V(T)$ a set of k sources. The* bottleneck vertex routing cost *of T, denoted by $C_V(T)$, is the maximum total distances from any vertex to all sources, i.e.,* $\max_{v \in V} D_T(v, S)$. *Given a graph and a set of k sources, the* bottleneck vertex routing cost spanning tree *(k-BVRT) problem is to find a spanning tree T with minimum $C_V(T)$ among all possible spanning trees.*

Definition 2. *Let T be a tree and a set of k sources $S \subset V(T)$. The* bottleneck source routing cost *of T, denoted by $C_S(T)$, is the maximum total distances from any source to all vertices, i.e.,* $\max_{s \in S} D_T(s, V)$. *Given a graph and a set of k sources, the* bottleneck source routing cost spanning tree *(k-BSRT) problem is to find a spanning tree T with minimum $C_S(T)$ among all possible spanning trees.*

Let G be a graph and $r \in V(G)$. A *shortest path tree* rooted at r is a spanning tree T of G such that $d_T(r, v) = d_G(r, v)$ for each vertex $v \in V$. That is, on a shortest path tree, the path from the root to any vertex is a shortest path in G. Let $R \subset V$. A forest F is a *shortest path forest* with multiple roots R if $d_F(v, R) = d_G(v, R)$ for any vertex $v \in V$, i.e., each vertex is connected to the closest root by a shortest path. A shortest path forest can be constructed by an algorithm similar to the shortest path tree algorithm. The time complexity is the same as the shortest path tree algorithm.

3 A PTAS for the 2-BVRT Problem

In this section, we show a PTAS for the 2-BVRT problem. Throughout this section, we assume that s_1 and s_2 are the two given sources. The PTAS is a modification of the one designed in [10] for the 2-MRCT. First we show that the simple 2-approximation algorithm for the 2-MRCT also works for the 2-BVRT problem. The 2-approximation algorithm first finds a shortest path between the two sources, and then constructs a shortest path forest with all the vertices of the path as the multiple roots. It returns the union of the path and the forest as the approximation solution. The following property is useful for showing the performance of the algorithm.

Lemma 1. *Let T be any spanning tree of $G = (V, E, w)$ and P be the path between s_1 and s_2 on T. $C_V(T) = w(P) + 2\max_{v \in V}\{d_T(v, P)\}$.*

Proof. First, $D_T(v,S) = d_T(v,s_1)+d_T(v,s_2) = w(P)+2d_T(v,P)$ for any $v \in V$. Since $C_V(T)$ is the maximum $D_T(v,S)$ over all vertices in V, the result follows. □

Now we show the performance of the simple algorithm.

Lemma 2. *A 2-approximation of the 2-BVRT can be found in $O(|V|\log|V| + |E|)$ time.*

Proof. First we show a lower bound of the optimum. Let Y and T be the optimal and approximation trees of the 2-BVRT problem. For any vertex $v \in V$, we have

$$C_V(Y) = \max_{u\in V}\{d_Y(u,s_1) + d_Y(u,s_2)\} \geq d_G(v,s_1) + d_G(v,s_2). \tag{1}$$

Since $d_G(s_1,s_2) \leq d_G(v,s_1) + d_G(v,s_2)$, we have $C_V(Y) \geq d_G(s_1,s_2)$. Together with Eq. (1), for any vertex v, we have $C_V(Y) \geq \frac{1}{2}(d_G(s_1,s_2) + d_G(v,s_1) + d_G(v,s_2))$. Since P is a shortest path between the two sources and each vertex is connected to P by a shortest path, for any vertex $v \in V$, we have $d_T(v,P) \leq \min\{d_G(v,s_1), d_G(v,s_2)\} \leq \frac{1}{2}(d_G(v,s_1) + d_G(v,s_2))$, and hence $2d_T(v,P) + d_T(s_1,s_2) \leq (d_G(v,s_1) + d_G(v,s_2)) + d_T(s_1,s_2) = (d_G(v,s_1) + d_G(v,s_2)) + d_G(s_1,s_2) \leq 2C_V(Y)$. By Lemma 1, we have $C_V(T) \leq 2C_V(Y)$ and T is a 2-approximation solution of the 2-BVRT. Note that the time-complexity is dominated by the construction of the shortest path forest. □

Now, we describe the PTAS and its performance analysis. By Lemma 1, it is easy to see that one can find the optimal if the path between the two sources is given. Similar to the previous PTAS for the 2-MRCT, our PTAS tries to guess some vertices of the path. For each guess, it first constructs a tree X spanning those guessed vertices, and then extends X to all other vertices by adding shortest paths to X. The performance is ensured by showing that at least one of the constructed trees is a good approximation solution and the approximation ratio approaches to 1 as the number of guessed vertices increasing. Although the algorithm is very similar to the previous one, the analysis of performance is different.

Let $\rho \geq 0$ be an integer parameter to be determined later. In the remaining paragraphs of this section, let Y be the optimal tree of the 2-BVRT. Also let $P = (p_1 = s_1, p_2, p_3, \ldots, p_h = s_2)$ be the path between the two sources on Y. The next lemma shows that we can choose only few vertices on P such that, for each vertex on P, there is a chosen vertex which is close enough to it. The lemma can be shown by induction on ρ but the proof is omitted in this abstract.

Lemma 3. *For any integer $\rho \geq 0$, there exists a subset $M \subset V(P)$ such that $|M| \leq \rho$ and $d_Y(v, M \cup \{s_1,s_2\}) \leq \frac{1}{\rho+2}w(P)$ for any $v \in V(P)$.*

The bound in the above lemma is tight. An extreme example is that the path have exactly $\rho+1$ internal vertices and all the edges are of the same length. Since we can only choose ρ vertices, there is one vertex left and the distance to its neighbor is $w(P)/(\rho+2)$.

Let M be a vertex set satisfying the property in Lemma 3 and $\bar{M} = M \cup \{s_1, s_2\}$. The next lemma can be shown by Lemma 3 but the proof is omitted here.

Lemma 4. *For any vertex v, $d_G(v, \bar{M}) \leq d_Y(v, P) + \frac{1}{\rho+2} w(P)$.*

By Lemma 4, we can design a good approximation algorithm for the 2-BVRT if the input graph is a metric graph. A metric graph is a complete graph with edge lengths satisfying the triangle inequality. In a metric graph, the edge between any pair of vertices is a shortest path. For a metric graph and an integer ρ, we try each possible path $(s_1, m_1, m_2, \ldots, m_i, s_2)$ for $i \leq \rho$. There exists at least one such paths whose internal vertices are also on the path P of the optimal tree in the same order and satisfy the property in Lemma 4. Since the graph is a metric graph, the length of the path is no more than $w(P)$. Connecting all other vertices to the path by a shortest path forest, we may have a spanning tree and it is a $(\rho + 4)/(\rho + 2)$-approximation of the 2-BVRT (shown later). However, a problem is encountered when the input is a general graph instead of a metric graph. For a guessed i-tuple $(m_1, m_2, \ldots, m_i)$ of vertices, there is no obvious way to construct such a desired path. We overcome the difficulty by a technique developed in [10] for 2-MRCT.

Lemma 5. *Suppose that P is a path on a general graph. Let s_1 and s_2 be the two endpoints and m_1, m_2,... ,m_i be i vertices such that P connects the consecutive m_j. Given the two endpoints of P and the i-tuple $(m_1, m_2, \ldots, m_i)$, there is an $O(in^2)$ time algorithm which constructs a tree X spanning the given vertices and having the property that $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$.*

We list the PTAS below.

Algorithm PTAS-2-BVRT
Input: A graph $G = (V, E, w)$ and $s_1, s_2 \in V$, and an integer $\rho \geq 0$.
Output: A spanning tree T of G.
1: For each $i \leq \rho$ and i-tuple $(m_1, m_2, \ldots m_i)$ of vertices **do**
Find a tree X as in Lemma 5.
Find the shortest path forest spanning $V(G)$ with all vertices in $V(X)$ as roots.
Let T be the union of the forest and X.
end for
2: Output the tree T with minimum $C_V(T)$ among all constructed trees.

The performance of the PTAS is shown in the next lemma.

Lemma 6. *The algorithm returns a spanning tree T with $C_V(T) \leq (\frac{\rho+4}{\rho+2}) C_V(Y)$*

Proof. Since the algorithm tries all possible i-tuple of vertices for all $i \leq \rho$, it is sufficient to show the approximation ratio for the i-tuple satisfying the property in Lemma 4. Let M be the set of vertices in the i-tuple and $\bar{M} = M \cup \{s_1, s_2\}$. As in Lemma 5, let X be a tree spanning M and $d_X(v, s_1) + d_X(v, s_2) \leq w(P)$ for any $v \in V(X)$.

For each vertex v, since it is connected to X by a shortest path, there exists a vertex $x \in V(X)$ such that $d_T(v, s_1) = d_G(v, x) + d_T(x, s_1)$ and $d_T(v, s_2) = d_G(v, x) + d_T(x, s_2)$. Therefore $D_T(v, S) = 2d_G(v, X) + d_T(x, s_1) + d_T(x, s_2)$. By Lemma 5, we have $D_T(v, S) \leq 2d_G(v, X) + w(P)$. Since $\bar{M} \subset V(X)$, $d_G(v, X) \leq d_G(v, \bar{M})$, and then by Lemma 4 we have $d_G(v, X) \leq d_Y(v, P) + \frac{1}{\rho+2} w(P)$. Consequently we have, for any vertex v,

$$D_T(v, S) \leq 2d_Y(v, P) + \frac{2}{\rho+2} w(P) + w(P) \leq \left(1 + \frac{2}{\rho+2}\right) D_Y(v, S),$$

since $w(P) \leq D_Y(v, S)$. Then the result is obtained by definition. □

The result of this section is summarized in the following theorem. The theorem can be easily shown by taking $\rho = \lceil \frac{2}{\varepsilon} \rceil - 2$ in the above lemma and the proof is omitted here.

Theorem 1. *The 2-BVRT problem admits a PTAS. For any constant $\varepsilon > 0$, a (1+ε)- approximation algorithm of the 2-BVRT of a graph G can be found in $O(n^{\lceil \frac{2}{\varepsilon} \rceil})$ time.*

4 The 2-BSRT Problem

In the section, we discuss the 2-BSRT problem. In the 2-BSRT problem, we are given a graph $G = (V, E, w)$ with two source vertices and asked to find a spanning tree. The objective function of the problem is $C_S(T) = \max_{s \in S} \{\sum_{v \in V} d_T(s, v)\}$. First we shall consider the case that the input is a general graph, and give a $(2 + \varepsilon)$-approximation algorithm for any fixed $\varepsilon > 0$. Then we show that the problem admits a PTAS if the input is restricted to a metric graph. Throughout this section, we assume that s_1 and s_2 are the two given sources.

4.1 On General Graphs

First we present a $(2+\varepsilon)$-approximation algorithm on general graphs. The algorithm is basically the same as algorithm **PTAS-2-BVRT** except that, among all the constructed trees, it returns the tree T with minimum $C_S(T)$ instead of $C_V(T)$. The main idea is similar to the PTAS for the 2-BVRT, but we need a different analysis of the performance. First we establish a lower bound of the cost in the next lemma. But omit the proof in this abstract.

Lemma 7. *Let T be a spanning tree of $G = (V, E, w)$ and P be the path between s_1 and s_2 on T. $C_S(T) \geq \frac{n}{2} w(P) + \sum_{v \in V} d_T(v, P)$.*

In the remaining paragraphs of this section, we shall use the following notations. Let Y be the optimal tree of the 2-BSRT. Also let $P = (p_1 = s_1, p_2, p_3, \ldots, p_h = s_2)$ be the path between s_1 and s_2 on Y. Now we introduce a partition of the vertices, which appeared in the previous work for the 2-MRCT problem (Fig. 1).

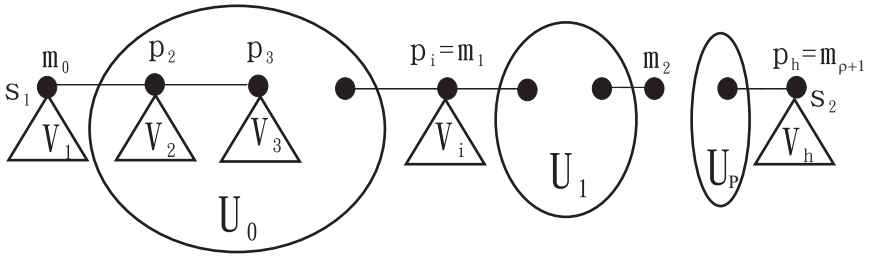


Fig. 1. The definitions of the partition of the vertices.

Define V_i, $1 \le i \le h$, as the set of the vertices connected to P at p_i and also let $p_i \in V_i$. Let $\rho \ge 0$ be an integer parameter to be determined later. For $0 \le i \le \rho + 1$, define $m_i = p_j$ in which j is the minimal index such that $\sum_{1 \le q \le j} |V_q| \ge \lceil i \frac{n}{\rho+1} \rceil$. By definition, $s_1 = m_0$ and $s_2 = m_{\rho+1}$. For $0 \le i \le \rho$, let $U_i = \bigcup_{a<j<b} V_j$ and P_i be the path from p_a and p_b, in which $p_a = m_i$ and $p_b = m_{i+1}$. Also let $U = V \setminus \bigcup_{0 \le i \le \rho} U_i$ and $M = \{m_i : 0 \le i \le \rho + 1\}$. Note that the above definitions include the case $m_i = m_{i+1}$. In such a case, P_i contains only one vertex and U_i is empty. By the above definitions, the vertex set V is partitioned into $U, U_0, U_1, \ldots, U_\rho$ satisfying the following properties. The lemma comes from [10]. We state the result but the proof is omitted.

Lemma 8. $\sum_{v \in U_i} d_G(v, M) \le \sum_{v \in U_i} d_Y(v, P) + \frac{n}{2(\rho+1)} w(P_i)$, *for any* $0 \le i \le \rho$ *and* $\sum_{v \in U} d_G(v, M) \le \sum_{v \in U} d_Y(v, P)$.

Our algorithm tries to correctly guess $m_1, m_2, \ldots m_\rho$ and construct a tree X spanning these vertices with the property that $d_X(v, s_1) + d_X(v, s_2) \le w(P)$ for any $v \in V(X)$. Once such a tree X has been constructed, we extend it to a spanning tree T by adding the shortest path forest with vertices in $V(X)$ as the multiple roots. In the next lemma, we show that T is a $(2 + \frac{1}{\rho+1})$-approximation of the 2-BSRT.

Lemma 9. *Let X be a tree spanning M and $d_X(v, s_1) + d_X(v, s_2) \le w(P)$ for any $v \in V(X)$. The spanning tree T, which is the union of X and the shortest path forest with vertices in $V(X)$ as the multiple roots, is a $(2 + \frac{1}{\rho+1})$-approximation solution of the 2-BSRT problem.*

Proof. (Outline) By definition, the vertices $m_1, m_2, \ldots, m_\rho$ partition the vertex set V into $(U, U_0, U_1, \ldots, U_\rho)$ and the subsets satisfy the properties in Lemma 8. Since $d_X(v, s_1) + d_X(v, s_2) \le w(P)$ for any $v \in V(X)$, we have

$$C_S(T) \le \sum_{v \in U} d_T(v, X) + \sum_{0 \le i \le \rho} \sum_{v \in U_i} d_T(v, X) + n\ w(P). \tag{2}$$

For $v \in U$, since $d_T(v, X) = d_G(v, X) \le d_G(v, M)$ and by Lemma 8, we have

$$\sum_{v \in U} d_T(v, X) \le \sum_{v \in U} d_G(v, M) \le \sum_{v \in U} d_Y(v, P). \tag{3}$$

Similarly $d_T(v, X) \leq d_G(v, M)$ for $v \in U_i$, and then by Lemma 8, we have

$$\sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_T(v, X) \leq \sum_{0 \leq i \leq \rho} \sum_{v \in U_i} d_Y(v, P) + \frac{n}{2(\rho+1)} w(P). \tag{4}$$

By Eqs. (2)–(4) and Lemma 7, we can obtain

$$C_S(T) \leq C_S(Y) + \left(\frac{\rho+2}{\rho+1}\right) C_S(Y) = \left(2 + \frac{1}{\rho+1}\right) C_S(Y).$$

□

Theorem 2. *For any constant $\varepsilon > 0$, there is an algorithm finding a $(2+\varepsilon)$-approximation of the 2-BSRT of a general graph in $O(n^{\lceil \frac{1}{\varepsilon}+1 \rceil})$ time.*

4.2 On Metric Graphs

Although the 2-BVRT problem and the 2-BSRT problem seem very similar, they are actually different. In the 2-BVRT problem, by Lemma 1, if the path between the two sources is given, the optimal tree can be easily obtained by connecting all other vertices to their closest vertices on the path. However, it does not hold for the 2-BSRT problem. There is no obvious way to find an optimal even the path is given. For this reason, the 2-BSRT seems more difficult to be approximated than the 2-BVRT. In the last section, we show how to approximate the 2-BSRT on general graphs within approximation ratio $2+\varepsilon$. We shall show that the 2-BSRT admits a PTAS if the input graph is restricted to a metric graph. The key point is that we employ an algorithm to find how to connect vertices to the vertices of the path so as to minimize the cost. The algorithm runs in polynomial time if the number of vertices of the path is constant.

The main idea of the PTAS is similar to the algorithm for the problem on general graphs. The definitions of Y, P, P_i, V_i, M, U, $U_0, U_1 \ldots U_\rho$ are also the same as in the previous section. To show the performance, we first construct a spanning tree $\bar{Y}$ from the optimal tree Y as follows. Note that the tree $\bar{Y}$ does not appear in the PTAS. It is used to show the performance of our PTAS.

- Let $\bar{P}$ be the path $(s_1 = m_0, m_1, m_2, \ldots, m_{\rho+1} = s_2)$. Since the graph is a metric graph, the path exists.
- For each $v \in V_j \subset U$ ($p_j \in M$), v is connected to p_j, i.e., edge $(v, p_j) \in E(\bar{Y})$.
- For each $v \in U_i$, v is connected to either m_i or m_{i+1} depended on which is closer to v on Y, i.e., edge $(m_i, v) \in E(\bar{Y})$ if $d_Y(v, m_i) \leq d_Y(v, m_{i+1})$, and $(m_{i+1}, v) \in E(\bar{Y})$ otherwise. Note that decision is made by the distances on Y but not on the original graph.

We shall show that $\bar{Y}$ is a $(\frac{\rho+3}{\rho+1})$-approximation of the 2-BSRT Y.

Lemma 10. *For $s \in \{s_1, s_2\}$ and $0 \leq i \leq \rho$, $D_{\bar{Y}}(s, U_i) \leq D_Y(s, U_i) + \frac{n}{(\rho+1)} w(P_i)$*

Proof. We show the result for s_1. The case of s_2 can be shown similarly. First, since the graph is a metric graph, $d_{\bar{Y}}(s_1, m) \leq d_Y(s_1, m)$ for any $m \in M$. For each $v \in V_j \subset U_i$, v is connected to either m_i or m_{i+1}. When v is connected to m_i, it is clear that $d_{\bar{Y}}(v, s_1) \leq d_Y(v, s_1)$ since the path in Y is just replaced with consecutive short-cut edges. Now we consider the case that v is connected to m_{i+1}. It only happens when $d_Y(v, m_{i+1}) < d_Y(v, m_i)$. Since $d_Y(v, m_i) = d_Y(v, p_j) + d_Y(p_j, m_i)$ and $d_Y(v, m_{i+1}) = d_Y(v, p_j) + d_Y(p_j, m_{i+1})$, the condition implies $d_Y(p_j, m_{i+1}) < d_Y(p_j, m_i)$, or equivalently $d_Y(p_j, m_{i+1}) < w(P_i)/2$. We have

$$\begin{aligned} d_{\bar{Y}}(v, s_1) &= d_{\bar{Y}}(s_1, m_{i+1}) + w(m_i, v) \\ &\leq ((d_Y(s_1, p_j) + d_Y(p_j, m_{i+1})) + (d_Y(v, p_j) + d_Y(p_j, m_{i+1})) \\ &= d_Y(v, s_1) + 2d_Y(p_j, m_{i+1}) \leq d_Y(v, s_1) + w(P_i). \end{aligned}$$

Consequently $d_{\bar{Y}}(v, s_1) \leq d_Y(v, s_1) + w(P_i)$ in both the cases. Since $|U_i| \leq \frac{n}{\rho+1}$, we have

$$\begin{aligned} D_{\bar{Y}}(s_1, U_i) &\leq \sum_{v \in U_i} (d_Y(v, s_1) + w(P_i)) \leq \sum_{v \in U_i} d_Y(v, s_1) + \frac{n}{(\rho+1)} w(P_i) \\ &= D_Y(s_1, U_i) + \frac{n}{(\rho+1)} w(P_i). \end{aligned}$$

□

The next lemma gives the approximation ratio of $\bar{Y}$, which can be shown by Lemma 10 and Lemma 7.

Lemma 11. *The spanning tree $\bar{Y}$ is a $(\frac{\rho+3}{\rho+1})$-approximation solution of Y.*

Proof. (Outline) For any vertex $v \in V_j \subset U$, it is connected to $p_j \in M$ in $\bar{Y}$, and $d_{\bar{Y}}(v, s_1) \leq d_Y(v, s_1)$. By Lemma 10, $D_{\bar{Y}}(s_1, V) \leq D_Y(s_1, V) + \frac{n}{(\rho+1)} w(P)$. Similarly $D_{\bar{Y}}(s_2, V) \leq D_Y(s_2, V) + \frac{n}{(\rho+1)} w(P)$. By definition, we have $C_S(\bar{Y}) = \max\{D_{\bar{Y}}(s_1, V), D_{\bar{Y}}(s_2, V)\} \leq C_S(Y) + \frac{n}{(\rho+1)} w(P) \leq (1 + \frac{2}{\rho+1}) C_S(Y)$, since $C_S(Y) \geq \frac{n}{2} w(P)$ by Lemma 7. □

We have transformed a 2-BSRT Y into a tree $\bar{Y}$ and have shown that $\bar{Y}$ is a good approximation of Y. Since the input is a metric graph, we may assume that the vertices of P are the only possible internal vertices of Y. As a result, the tree $\bar{Y}$ belongs to a special kind of spanning trees in which there are only $(\rho + 2)$, or less, internal vertices and the internal vertices form a path with the two sources as the endpoints. Our PTAS for the 2-BSRT is designed to find the best spanning tree of the kind.

Definition 3. *A λ-path is a path between s_1 and s_2 and containing exactly $(\lambda + 2)$ vertices.*

Definition 4. *A spanning tree T is a λ-path tree if $SP_T(s_1, s_2)$ is a λ-path and all the internal vertices are on the path.*

By the definition and Lemma 11, we have the next corollary.

Corollary 1. *For a metric graph and any integer $\rho \geq 0$, there exists a λ-path tree, $\lambda \leq \rho$, which is a $(\frac{\rho+3}{\rho+1})$-approximation of the 2-BSRT.*

Our PTAS finds the λ-path tree with minimum bottleneck source routing cost for each $\lambda \leq \rho$, and returns the best of them. By Corollary 1, it is a $(\frac{\rho+3}{\rho+1})$-approximation. Now we show how to find the best λ-path tree of a metric graph for a fixed λ. The idea is from [9], which is used to find the minimum routing cost tree with constant number of internal nodes.

A λ-path tree can be described by a λ-path $Q = (s_1 = q_0, q_1, \ldots, q_{\lambda+1} = s_2)$ and a partition $\mathcal{L} = (L_0, L_1, \ldots, L_{\lambda+1})$ of $\bar{V}$, in which $\bar{V} = V \setminus V(Q)$ and L_i is the set of vertices in $\bar{V}$ and adjacent to q_i. We denote a λ-tree by $(Q, \mathcal{L})$. Let $l = (l_0, l_1, \ldots . l_{\lambda+1})$ be a nonnegative $(\lambda+2)$-vector such that $\sum_{i=0}^{\lambda+1} l_i = n-\lambda-2$. We say that a λ-path tree $(Q, \mathcal{L})$ has the configuration (Q, l) if $l_i = |L_i|$ for all $0 \leq i \leq \lambda + 1$. We define $\alpha(Q, l)$ to be the λ-path tree of minimum bottleneck source routing cost with a configuration (Q, l). We shall show that $\alpha(Q, l)$ can be found in polynomial time.

Lemma 12. *For a given configuration (Q, l), the optimal λ-path tree $\alpha(Q, l)$ can be found in $O(n^3)$ time.*

Proof. (outline) Let $T = (Q, \mathcal{L})$ be a λ-path spanning tree with configuration (Q, l). We show that

$$D_T(s_1, V) = D_T(s_1, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_1, q_i) + \sum_{i=0}^{\lambda+1} \sum_{v \in L_i} w(v, q_i).$$

Similarly,

$$D_T(s_2, V) = D_T(s_2, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_2, q_i) + \sum_{i=0}^{\lambda+1} \sum_{v \in L_i} w(v, q_i).$$

Define $f_1(Q, l) = D_T(s_1, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_1, q_i)$ and $f_2(Q, l) = D_T(s_2, V(Q)) + \sum_{i=0}^{\lambda+1} l_i \; d_T(s_2, q_i)$. Note that the two function depend only on Q and l, and therefore two trees with the same configuration have the same values of the two functions. We can have $C_S(T) = \max\{f_1(Q, l), f_2(Q, l)\} + \sum_{i=0}^{\lambda+1} \sum_{v \in L_i} w(v, q_i)$. For a specified configuration (Q, l), the cost depends only on the second term. As a result, $\alpha(Q, l)$ can be constructed by finding a minimum-cost way of matching up the vertices of $\bar{V}$ to those in $V(Q)$ which obeys the degree constraints imposed by l. It is equivalent to find the minimum-cost perfect matching on the auxiliary complete bipartite graph H with vertex set $\bar{Q} \cup \bar{V}$ in which $\bar{Q}$ contains l_i copies of q_i for each i. Since the bipartite perfect matching problem, also called as the assignment problem, can be solved in $O(n^3)$ time [1], $\alpha(Q, l)$ can be solved with the same time complexity for a given configuration. □

For a λ-path Q, we may find the best λ-path tree by trying every possible nonnegative $(\lambda+2)$-vector l. Since there are $O(n^{\lambda+1})$ such vectors and each vector corresponds to one instance of the assignment problem, it takes $O(n^{\lambda+4})$ time if we solve the assignment problems individually. As in [9], by carefully ordering the assignment problems for the vectors and exploiting the common structure of two consecutive problems, we can obtain the optimal solution of every vector in this order by performing a single augmentation on the optimal solution of the previous vector. By this method, the optimal solution for each vector can be found in $O(\lambda n)$ time from the solution for its predecessor. We state the result in the next lemma but the proof is omitted in this abstract.

Lemma 13. *For any constant $\lambda \geq 0$, the λ-path tree of minimum bottleneck source routing cost can be found in $O(n^{2\lambda+2})$.*

In summary, we show the 2-BSRT problem on metric graphs admits a PTAS.

Theorem 3. *There exists a PTAS for the 2-BSRT problem on metric graphs, which finds a $(1+\varepsilon)$-approximation solution in $O(n^{\lceil 4/\varepsilon \rceil})$ time.*

5 The k-BVRT and the k-BSRT for $k > 2$

In this section we show that both the k-BVRT and the k-BSRT problems can be approximated with ratio 3 by a shortest path tree. First, we show the result for the k-BVRT problem.

Theorem 4. *Any shortest path tree rooted at any vertex is a 3-approximation of the k-BVRT.*

Proof. Let Y be the optimal tree of the k-BVRT, we have

$$C_V(Y) = \max_{v \in V} \{D_Y(v,S)\} \geq \max_{v \in V} \{D_G(v,S)\}.$$

That is, $C_V(Y) \geq D_G(v,S)$ for any vertex v. Let T be any shortest path tree rooted at an arbitrary vertex r.

$$\begin{aligned} C_V(T) &= \max_{v \in V} \left\{ \sum_{s \in S} d_T(v,s) \right\} \leq \max_{v \in V} \left\{ \sum_{s \in S} (d_T(v,r) + d_T(r,s)) \right\} \\ &\leq \max_{v \in V} \left\{ \sum_{s \in S} ((d_G(v,s) + d_G(r,s)) + d_G(r,s)) \right\} \\ &= \max_{v \in V} \{D_G(v,S)\} + 2D_G(r,S) \leq 3C_V(Y). \end{aligned}$$

□

Corollary 2. *Given a general graph and a set of sources, a 3-approximation of the k-BVRT can be found in $O(|V| \log |V| + |E|)$ time.*

Similarly we have the next result for the k-BSRT problem. Note that the shortest path tree must be rooted at a source to ensure the performance, which is different from the k-BVRT problem. The proof is omitted in this abstract.

Theorem 5. *Any shortest path tree rooted at any source is a 3-approximation of the k-BSRT.*

Corollary 3. *Given a general graph and a set of sources, a 3-approximation of the k-BSRT can be found in* $O(|V| \log |V| + |E|)$ *time.*

6 Conclusion

In this paper, we investigate the k-BVRT and the k-BSRT problems and propose approximation algorithms. While the 2-BVRT problem having been shown to admit a PTAS, the 2-BSRT seems more difficult to approximate. We show a PTAS for the 2-BSRT only for metric graphs. For general graph, we only have a $(2 + \varepsilon)$-approximation algorithm. Improved approximation algorithms will be interesting. Another open problem left in the paper is the approximabilities of the two problems with more than two sources. The approximation algorithms in this paper are very simple, and better results are expected.

References

1. Ahuja, R.K., Magnanti, T.L., and Orlin, J.B.: Network Flows –Theory, Algorithms, and Applications. Prentice–Hall (1993).
2. Connamacher, H.S. and Proskurowski, A.: The complexity of minimizing certain cost metrics for k-source spanning trees. Discrete Applied Mathematics **131** (2003) 113–127.
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C.: Introduction to Algorithm. 2nd edition, MIT Press, Cambridge (2001).
4. Farley, A.M., Fragopoulou, P., Krumme, D.W., Proskurowski, A., and Richards, D.: Multi-source spanning tree problems. Journal of Interconnection Networks **1** (2000) 61–71.
5. Garey,M.R., and Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, San Francisco (1979).
6. Johnson, D.S., Lenstra, J.K., and Rinnooy Kan, A.H.G.: The complexity of the network design problem. Networks **8** (1978) 279–285.
7. Thorup, M.: Undirected single source shortest paths with positive integer weights in linear time. Journal of ACM **46** (1999) 362–394.
8. Wu, B.Y., Chao, K.M., and Tang, C.Y.: Approximation algorithms for some optimum communication spanning tree problems. Discrete Applied Mathematics **102** (2000) 245–266.
9. Wu, B.Y., Lancia, G., Bafna, V., Chao, K.M., Ravi, R., and Tang, C.Y.: A polynomial time approximation scheme for minimum routing cost spanning trees. SIAM Journal on Computing **29** (2000) 761–778.
10. Wu, B.Y.: A polynomial time approximation scheme for the two-source minimum routing cost spanning trees. Journal of Algorithm **44** (2002) 359–378.
11. Wu, B.Y.: Approximation algorithms for the optimal p-source communication spanning tree, to appear in Discrete Applied Mathematics.

Efficient Sequential and Parallel Algorithms for Popularity Computation on the World Wide Web with Applications against Spamming

Sung-Ryul Kim*

Division of Internet and Media
& Center for Aerospace System Integration Technology
Konkuk University, Seoul, Korea

Abstract. When searching for information on the World Wide Web, it is often necessary to use one of the available search engines. Because the number of results are quite large for most queries, it is very important to have some measure of relevance of the result Web pages. One of the most important relevance factors is the popularity score which indicates how popular the page is among the users. We propose a modified version of status index by Katz and present efficient sequential and parallel algorithms that solve the problem. The high flexibility in our algorithm results in resilience to spamming as we show by experiments.

1 Introduction

As the World Wide Web (or the Web for short) is becoming bigger and bigger, it is absolutely necessary to use some kind of search capability to find information on the Web. These days, many *search engines* are available that enable us to find the information that we seek. Typically, the query to a search engine consists of just a few keywords and the search engine finds the Web pages that contain all of the given keywords. Because the number of keywords in the query is small, there are tremendous number of results unless the query contains a very specific combination of keywords.

In many cases, what the user wants is a small set of pages that are relevant to what he has in mind, not just any page that contains all of the keywords that he has given to the search engine. For example, when the query is "apple computer," what the user intended is most likely the Apple Computer site. However, many other pages also contain both keywords and become legitimate (but irrelevant) results. If the results are given without any ordering, then the results become useless to the user.

So the issue for the search engine is to find the relevant pages and show the relevant ones first. Many heuristics are used to compute the relevance of a page. One is the use of the content of a page and anchor text, i.e., the text that appear close to the link to the page from some other page. Some examples are

* Supported by Korea Research Foundation Grant KRF-2002-003-D00304

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 367–375, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the relative frequency of the keywords, the location of keywords such as being in the title or appearing close to the start of a page, and the proximity of keywords, i.e., how close the keywords appear together in a page [9].

Also, there are models that use the link structure of the Web to determine the relative importance (or popularity) of the pages and use the score as a factor in the ranking. A simple (but not very useful) example is the method of counting *backlinks* that comes into a page. A backlink is a link from some other page which points to the page. Other examples with better results include the *hub and authority* model by Kleinberg [7], the *PageRank* model by Page, et al. [3, 8], and the *status index* method by Katz [6], which is a generalization of the backlink counting method. There are also a few similar methods available in the literature [4,2]. We focus on the status index method in this paper.

1.1 Related Works

In the *hub and authority* model, hubs and authorities are mutually defined. That is, a hub is a page with has many links to authorities and an authority is a page which is pointed to by many hubs. This heuristic is based on the intuition that a page with high authority score is more likely to be relevant than a page with low authority score.

In the PageRank model the rank $R(v_i)$ of a page v_i is defined as

$$R(v_i) = d \sum \frac{R(v_j)}{h_j} + (1 - d),$$

where the sum ranges over all pages v_j that has a link to v_i, h_j is the number of outgoing links from v_j, and d is the *dumping factor*. The PageRank model can be considered to be a random walk model. That is, PageRank of a page v_i is the probability that a random walker (which continues to follow arbitrary links to move from page to page) will be at v_i at any given time. The dumping factor corresponds to the probability of the random walk to jump to an arbitrary page, rather than to follow a link, on the Web. It is required to reduce the effects on the PageRank computation of loops and dangling links in the Web.

In the status index method, which is a generalization of the backlink-counting method, the status index of a page is determined by the number of directed paths that ends in the page, where the influence of longer paths is attenuated by a decay factor. The length of a path is defined to be the number of edges it contains. The status index $I(v_i)$ of a page v_i is formally defined as follows.

$$I(v_i) = \sum_{k=0}^{\infty} [\alpha^k N(v_i, k)]$$

where $N(v_i, k)$ is the number of paths of length k that starts at *any* page and ends at v_i and α is the decay factor. Solutions for all the pages are guaranteed to exist as long as α is smaller than λ^{-1}, where λ is the maximum in-degree of any page.

For all three methods above, it is not practical to compute the exact solution for a large database. Instead, various schemes are used to compute an approximate solution.

1.2 Our Results

We propose a variation of the status index by Katz and develop efficient sequential and parallel algorithms to compute the scores. Our modifications to the definition will also give a great flexibility in computing the status index.

Flexibility is needed because it is possible to influence the computation of popularity score in an adverse way to obtain higher score. The *spamming* usually involves generating pages containing many links that are otherwise unused. These pages are used to boost, for example, the PageRank of a target page. There exists companies that claim to be able to boost the popularity score of Web pages of their clients. Search engine companies are trying to reduce the effects of spamming by banning the sites that are suspected of spamming. The sites to be banned are usually selected manually and thus it requires a lot of human labor. Some companies are even promising rewards to people who report spamming sites to them.

Our modifications include the ability to designate a set of pages that are known to be popular as the only possible sources of paths, to set different weights to each source, and to freely control the decay factor for each path length. These modifications will lead to more variations in the computation of status index that can be used to reduce the effects of spamming. We show that our algorithm is highly resilient to spamming by implementing our algorithm and comparing the results to that of PageRank. We also show that the manual work required to use our algorithm is very small compared to manual banning of spamming sites.

2 Definitions and Preliminary Properties

The input to the algorithms is a triple $G = (V, E, S)$ where V is the set of all pages in the Web, E is the set of links, and S is a subset of V called the source set. For each node s in S, a weight W_s is also given. As noted before, the nodes in S consists of known popular pages on the Web. The set S and the weights of the nodes in S is assumed to be set manually. We later show that it is enough to have only a few tens of pages in S to compute a popularity score comparable to PageRank. The pages and links will be called *nodes* and *edges* in the description of the algorithms.

Our definition of the status index $I(v_i)$ of a node v_i is as follows.

$$I(v_i) = \sum_{k=0}^{l} [f(k) \sum_{s \in S} W_s N(v_i, k, s)]$$

where $N(v_i, k, s)$ is the number of paths of length k that starts at node $s \in S$, $f(k)$ is the decay factor for length k, and l is the limit of path length. The

differences from the status index by Katz is as follows. Firstly, the start nodes of the paths are restricted to those in S. Secondly, each source node s is given a weight W_s. Finally, the decay factor α is replaced by a function $f(k)$. These modifications will result in a greater resilience against spamming, as we show by experiments later.

It is assumed that $f(k)$ decreases faster than $(1/(\lambda + 1))^k$ where λ is the maximum out-degree of the nodes in V. It can be easily shown that $\sum_{v_i}[f(k)\sum_{s\in S} W_s N(v_i, k, s)]$ where v_i ranges over all nodes in V becomes exponentially smaller as k increases. Thus, we can guarantee that the result is an approximation of the summation to the infinity.

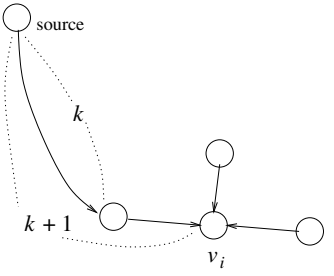


Fig. 1. The last edge

Lemma 1. *If $w_1, w_2, \ldots, w_n$ are the nodes that have edges to v_i, source node s is in S, and the length of path $k > 0$ then*

$$N(v_i, k, s) = \sum_{j=1}^{n} N(w_j, k-1, s).$$

Proof. It is obvious that the last edge of any path of length k from s to v_i is an edge from a node w_j $(1 \le j \le n)$ to v_i. By eliminating the last edge from the path, we find a path of length $k-1$ from s to w_j. Conversely, if we have a path of length $k-1$ from s to a w_j, we find a path of length k from s to v_i by adding (w_j, v_i) at the end of the path. Thus, we have found a one-to-one correspondence.

3 Sequential Algorithm

The sequential algorithm works in $l+1$ rounds numbered from zero to l. In each round k we compute $\sum_{s\in S} W_s N(v_i, k, s)$ for each node v_i. Note that it is enough to compute the final solution for v_i.

In round zero, we have to compute $\sum_{s\in S} W_s N(v_i, 0, s)$ for each node v_i. If $v_i \notin S$, then there are no paths of length zero from a node in S to v_i. Thus the value to be computed for v_i is zero. If $v_i \in S$, then there is only one path, from v_i to v_i, of length zero from a node in S to v_i. So the value to be computed for v_i is W_{v_i}.

In round $k > 0$, we have to compute $\sum_{s \in S} W_s N(v_i, k, s)$ for each node v_i. Let $w_1, w_2, \ldots, w_n$ be the nodes that have edges to v_i. We know that

$$\sum_{s \in S} W_s N(v_i, k, s) = \sum_{s \in S} W_s \sum_{j=1}^{n} N(w_j, k-1, s)$$

If we receive $N(w_j, k-1, s)$ for each source node s separately from each node w_j, the time complexity will depend linearly on $|S|$, which may be very large because we do not set any bounds on the number of sources. Further, we have to have a memory of size $|S|$ for each node. However, it is possible to avoid the problems by rearranging the formula slightly as follows.

$$\begin{aligned}\sum_{s \in S} W_s N(v_i, k, s) &= \sum_{s \in S} W_s \sum_{j=1}^{n} N(w_j, k-1, s) \\ &= \sum_{s \in S} \sum_{j=1}^{n} W_s N(w_j, k-1, s) \\ &= \sum_{j=1}^{n} \sum_{s \in S} W_s N(w_j, k-1, s)\end{aligned}$$

Because $\sum_{s \in S} W_s N(w_j, k-1, s)$ is computed for each node w_j in the previous round, we can compute $\sum_{s \in S} W_s N(v_i, k, s)$ for v_i by receiving one value for each incoming edge. The computed sum (after multiplied by $f(k)$) can be added to the variable that will hold the final score for each node.

Theorem 1. *Given $G = (V, E, S)$ where V is the set of all pages in the Web, E is the set of links, and S is a subset of V and weight W_s for each $s \in S$, the modified status index can be computed in $O(l(|E| + |V|))$ time.*

4 BSP Algorithm

BSP is a practical parallel computation model that can be applied to distributed computing environments. The detailed description of the BSP algorithm will appear in the full version of the paper.

5 Robustness

We say that a computation of popularity score is *robust* if it is not affected very much by a malicious modification of Web pages. However, popularity is a subjective concept and it is very difficult to define a measure of robustness. For example, if the spammer can modify arbitrary pages in the Web, then no computation can distinguish real popular pages from other pages using only the structure of links.

Thus we first restrict the pages that a spammer is able to modify. There are many sites on the Web that are very popular and well known to hold authoritative information. We assume that the spammer cannot modify the links appearing on those authoritative sites. We also assume that the spammer cannot modify the links on the pages that have short paths (say, of length at most 5) from the pages in the authoritative sites. These restrictions appeals to the intuition that spammers will not have links from authoritative pages, say the pages in http://www.yahoo.com/, to their pages. The same intuition leads us to the fact that a page is more likely to be a spamming page if it is far away from an authoritative page. The ability to freely control the decay factor becomes important from this respect. Thus we may make $f(k)$ decrease much faster if k is bigger than a certain value.

6 Experimental Results

We have designed the experiment with two objectives. One is to show that the flexibility of our algorithm can be used to make the computation robust. The other is to show that the quality of the result is as good as PageRank. For those objectives, we have implemented the PageRank algorithm and three versions of our algorithm, with varying degrees of robustness.

We have tested the four algorithms on sets of data we have constructed to simulate the Web with varying degrees of spamming. The data set we have constructed consists of two kinds of pages.

- Regular pages: Regular pages corresponds to the normal pages on the Web. Because the pages in a site usually form a structure similar to a tree, we have generated sets of pages of varying sizes and we have built the pages in a set into a tree structure. Thus, each page in a set has as links to its children, parent, ancestors, and its siblings. In addition, each page has randomly generated links to pre-selected popular pages in the whole set of regular pages. Each page that is selected as a popular page is given a probability that it will receive an incoming link from other pages. The existence of the popular pages in our experiment corresponds to real popular pages on the Web. Each regular page has from 5 to 20 outgoing links.
- Spamming pages: The spamming pages are constructed so that they will simulate the *promotion* techniques used by link spammers. The spamming pages are tightly cross-linked and they have links to a few designated pages that are the targets of spamming. Each spamming page has 20 outgoing links.

Finally, the data set is constructed so that there are a few links from regular pages to spamming pages. Usually, the spammers cannot get links from very popular pages. Instead, they get links from many less popular pages. In our data set, the links from regular pages to spamming pages originate from pages that are in lower parts of the trees. In that way, we can be sure that there will not be links from popular pages to spamming pages.

The popularity-computation methods tested are as follows.

- **PageRank** implements the PageRank method described in Section 1. The dumping factor d is set at 0.2.
- **Katz, All Source** implements our algorithm where the source set S contains all (regular and spamming) pages. The weights to the source pages are set to be the same values. The decay factor $f(k)$ is defined as $1/(20+1)^k$ because the maximum number of outgoing links from a page is 20.
- **Katz, Selected Source** implements our algorithm where the source set S contains a few of the pre-selected popular pages. Only the pages with very high probabilities are in S. The selection of the pages in S has to be performed manually. So it is very important that the number of pages in S is small. The decay factor is defined in the same way as **Katz, All Source**.
- **Katz, Cutoff** also implements our algorithm. The source set S is selected in the same way as in **Katz, Selected Source**. However, the decay factor is defined as $1/(20+1)^k$ if $k < 10$, $1/40^k$ otherwise. By this definition, longer paths will contribute much less than it contributes in the two previous implementations.

In the following we present the results of the comparison.

6.1 Robustness

We compare the robustness of the four implementations to spamming. For the testing we have generated 1,000,000 regular pages. The source set S for **Katz, Selected Source** and **Katz, Cutoff** consisted of 20 regular pages that have the highest probability of receiving random links from other pages in the generation of the test data. That is, they are the known popular pages in the test data. To simulate varying degrees of spamming, we have constructed two kinds of data sets. In the first kind of data sets, the spamming pages amount to 10% of regular pages. In the second kind of data set (called 500% spamming case), there are five time more spamming pages than regular pages.

For each kind of data set, we have randomly built 10 data sets and tested the four implementations. After each computation is complete, we have sorted the pages by the scores computed by an implementation and counted the number of spamming pages that are ranked from first to 10-th, 11-th to 100-th, 101-st to 1000-th, and etc. In the tables, the column titled as 10^k means the pages ranked $10^{k-1}+1$-st to 10^k-th by each implementation.

Table 1 shows the average results of the four implementations on 10 data sets of 10% spamming case. From the result we can see that **PageRank** show some effects of spamming as 10.0% of pages ranked from 11-th to 100-th are spamming pages. It is noticeable that **Katz, All Source** is much less robust than **PageRank**. In PageRank, as the number of outgoing links from a page increases, the contribution made by each link becomes smaller. But in our algorithm, the contribution of a link is the same regardless of the number of outgoing links from a page. Since spamming pages have more outgoing links on the average

Table 1. Ratio of spamming pages in 10% spamming case

	Rank range					
Method	10	10^2	10^3	10^4	10^5	10^6
PageRank	0%	10.0%	16.7%	10.4%	10.8%	9.9%
Katz, All Source	90.0%	72.2%	19.8%	43.9%	94.0%	1.2%
Katz, Selected Source	0.0%	0.0%	0.0%	1.8%	3.0%	10.8%
Katz, Cutoff	0.0%	0.0%	0.0%	0.1%	0.3%	0.8%

Table 2. Ratio of spamming pages in 500% spamming case

	Rank range					
Method	10	10^2	10^3	10^4	10^5	10^6
PageRank	92.0%	93.2%	76.8%	50.6%	46.2%	82.7%
Katz, All Source	99.0%	98.9%	98.0%	97.1%	94.4%	99.5%
Katz, Selected Source	0.0%	0.0%	0.0%	1.8%	2.6%	14.9%
Katz, Cutoff	0.0%	0.0%	0.0%	0.1%	0.3%	4.8%

than regular pages, the effect of the previous observation is shown in the result. However, it is obvious that both **Katz, Selected Source** and **Katz, Cutoff** is a lot more robust than both **PageRank** and **Katz, All Source**.

Table 2 shows the results of testing on the 500% spamming case. We can see that the effect of the increase in the number of spamming pages is obvious in both **PageRank** and **Katz, All Source**. However, we can see that both **Katz, Selected Source** and **Katz, Cutoff** are not affected by the increase in the number of spamming pages by much. Also, we can see that **Katz, Cutoff** is more robust than **Katz, Selected Source**.

6.2 Ranking Quality

The experimental results for ranking quality appears in the full version of the paper. The experiments show that the result of our algorithm is very similar to that by PageRank.

7 Conclusion

In this paper we proposed a modified version of the status index by Katz and presented efficient sequential and parallel algorithms that solve the problem. We argued that because of the flexibility that our definition of the status index gives to the search engine, it is highly likely that the status index computation is robust to spamming. We have provided the evidence by comparing the robustness of actual implementations of PageRank and variations of our algorithm. The experimental results show that the quality of ranking for normal pages is comparable to PageRank. We have also shown that only a little manual handling is required to use our algorithm.

References

1. Berenbrink, P., Meyer auf der Heide, F., Schröder, K., Allocating weighted jobs in parallel, *Theory of Computing Systems*, 32, 281–300, 1999.
2. Bonacich, P., Lloyd, P., Eigenvector-like measures of centrality for asymmetric relations, *manuscript*.
3. Brin, S., and Page, L., The anatomy of a large-scale hypertextual web search engine, *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.
4. Hubbell, C. H., An input-output approach to clique identification, *Sociometry*, 28, 377–399, 1965.
5. J. JáJá, *An Introduction to Parallel Algorithms*,
6. Katz, L., A new status index derived from sociometric analysis, *Psychometrika*, 18, 39–43, 1953.
7. Kleinberg, J., Authoritative sources in a hyperlinked environment, *Journal of the ACM*, 46(5), 604–632, 1999.
8. Page, L., Brin, S., Motwani, R., and Winograd, T., The PageRank citation ranking: Bringing order to the Web, *Technical report, Stanford University*, 1998. Addison-Wesley, 1992.
9. Sadakane, K., and Imai, H., Fast algorithms for k-word proximity search, *IEICE Trans. Fundamentals*, Vol. E84-A, No.9, 312–319, Sep. 2001.
10. L. G. Valiant, A bridging model for parallel computation, *Comm. ACM*, 33:103–111, 1990.
11. L. G. Valiant, General purpose parallel architectures, In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pp. 943–972, Elsevier/The MIT Press, Amsterdam, 1990.

Decentralized Inter-agent Message Forwarding Protocols for Mobile Agent Systems

JinHo Ahn

Dept. of Computer Science, Kyonggi University
San 94-6 Yiuidong, Yeongtonggu, Suwon Kyonggido 443-760, Republic of Korea
jhahn@kyonggi.ac.kr

Abstract. Mobile agent technology has emerged as a promising programming paradigm for developing highly dynamic and large-scale service-oriented computing middleware due to its desirable features. For this purpose, first of all, scalable and location-transparent agent communication issue should be addressed in mobile agent systems despite agent mobility. In this paper, we present efficient distributed directory service and message delivery protocols based on forwarding pointers to significantly reduce the length of chains of forwarding pointers by forcing the corresponding service host to maintain each mobile agent's pointer only after every k($k > 1$) migrations finished. This feature results in low message forwarding overhead and low storage and maintenance cost of increasing chains of pointers per host. Additionally, they enable each sending agent to communicate with mobile agents more faster than previous protocols by effectively using their location information in the sending agent's binding cache.

Keywords: Large-scale service oriented computing, mobile agent, directory service, message delivery, forwarding pointer

1 Introduction

Recently along with rapid advances in high speed backbone networks, portable devices such as cellular phones and personal digital assistants have been increasingly diffused in home and office computing environments and temporarily connected to Internet through wireless networks. According to this current technology trend, Internet service providers(ISPs) attempt to provide their services for users as follows[2]; not only traditional Internet services, but also newly developed services should be transparently and adaptively provided depending on dynamic logical or physical properties such as potable device characteristics, their locations and user. In order to satisfy these requirements, large-scale service oriented computing middleware is required to have new features such as runtime new services deployment, location awareness and context adaptation.

Mobile agent is an autonomous and self-contained program that moves between several nodes and executes its task on behalf of its user or application

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 376–385, 2004.
© Springer-Verlag Berlin Heidelberg 2004

program by using services supported on the nodes[3,8,11]. Due to these desirable features, i.e., dynamicity, asynchronicity and autonomy, mobile agent technology has emerged as a promising programming paradigm for developing the service-oriented computing middleware in various application fields such as e-commerce, telecommunication, DBMS, active networks and the like[9,12]. But, some important issues should be considered in order to fully exploit mobile agent capabilities. In this paper, we intend to focus on agent communication related issue, specifically distributed directory service and message delivery. Generally, this problem results from agent mobility because although messages are sent to locations where agents are known to be, the messages may not be delivered to the agents when they have moved. Thus, location-transparent communication services are needed to route and deliver messages correctly and automatically to mobile agents despite changing their locations. These communication service systems are essentially classified into two approaches, home agent based and forwarding pointer based. In the home agent based approach such as Aglets[4] each mobile agent is associated with a home agent host. In other words, whenever a mobile agent moves to a new service host, the agent should inform its home host of its new location. Thus, if a message has to be delivered to a mobile agent, the message must be first sent to its home agent, which forwards the message to the mobile agent. However, as the systems scale up, this behavior leads to home agent centralization, which may hamper the scalability of the global infrastructure. Moreover, when the home agent host is disconnected from the network, message delivery via home agents may be impossible.

Forwarding pointer based approach such as Voyager[7] enables each agent migrating to leave trails of forwarding pointers on hosts it has visited[5,6]. Thus, this approach can avoid performance bottlenecks of the global infrastructure, and therefore improve its scalability, particularly in large-scale distributed systems, compared with home based directory service approach. Additionally, even if a home host is disconnected from the rest of the network, the forwarding pointer based approach allows agents registering with the host to communicate with other agents. However, the approach has two practically important drawbacks[1]. First, if agents frequently migrate and therefore the length of chains of pointers increases, message forwarding overhead may not be negligible. In addition to this, if the system should serve a large number of mobile agents, the number of forwarding pointers each host should maintain in its storage may rapidly increase. To avoid an increasing cost of communication when agents frequently migrate, a previous work[5,6] pointed out that some techniques are needed to reduce the length of the chains of pointers, but didn't present concrete mechanisms to address the problems.

In this paper, we first present two scalable distributed directory service protocols based on forwarding pointers to solve the two problems stated above. Only after every k migrations have been completed, the protocols force the corresponding service host to maintain each mobile agent's pointer in its location table. This feature results in low message forwarding overhead and low storage and maintenance cost of increasing chains of pointers per host. Also, our proposed message delivery protocol allows each sending agent to communicate with

mobile agents more faster than the previous protocol[5]. For this purpose, it uses their location information in each sending agent's binding cache.

Due to space limitation, our system model, formal descriptions of the two proposed protocols and related work are all omitted. The interested reader can find them in [1].

2 Two Forwarding Pointer-Based Directory Service Protocols

This section introduces two protocols, Directory Service Protocol with Home Update($DSPHU$) and Directory Service Protocol with No Home Update($DSPNHU$). They consist of common and different parts respectively. In both protocols, after each agent migrates from its home host to the first visiting host x, the agent informs its current location only to x every migration until it has visited $(k-1)$ more hosts. In this case, x is called *Location Manager* of the agent. When the agent moves to the k-th visiting host y, the agent is registered at the new location manager y and then de-registered from the previous location manager x. At this point, the protocol $DSPHU$ forces x to inform the home of the agent that y is the location manager of the agent from now. But, the protocol $DSPNHU$ doesn't execute this update at the home. Due to their respective behaviors, the two protocols have different tradeoffs with respect to home update overhead and initial communication cost. Afterwards, x becomes *Forwarder* for the agent just having its forwarding pointer and y plays the role of the location manager for the agent while its $(k-1)$ more migrations have been performed. For both protocols, every host H_i should have the following four variables.

• $A_Locations_i$: a set for saving location information of every agent created in H_i. Its element is a tuple (*aid*, *c_hid*, *t_time*). *c_hid* for agent *aid* is the identifier of the host where H_i knows the agent is currently located and running. *t_time* is a timestamp associated with agent *aid* when the agent is located at host H_{c_hid}. It is necessarily required to avoid updating recent location information by older information[5].
• R_Agents_i: a set for saving location information of every mobile agent remotely created, but running on H_i. Its element is a tuple (*aid*, *l_hid*, *h_cnt*, *t_time*). *l_hid* is the identifier of the current location manager host of agent *aid*. Thus, when agent *aid* migrates to H_i, H_i should inform *l_hid* of the current location information of agent *aid*. *h_cnt* is a hop counter incremented every time the agent changes its location, but reset to one if the counter's value is more than the maximum number k. *t_time* is the timestamp associated with agent *aid* when migrating to H_i.
• $M_Locations_i$: a set for saving location information of every mobile agent which is not running on H_i, but whose location is currently managed by H_i. Its element is a tuple (*aid*, *c_hid*, *t_time*). *c_hid* is the identifier of the current

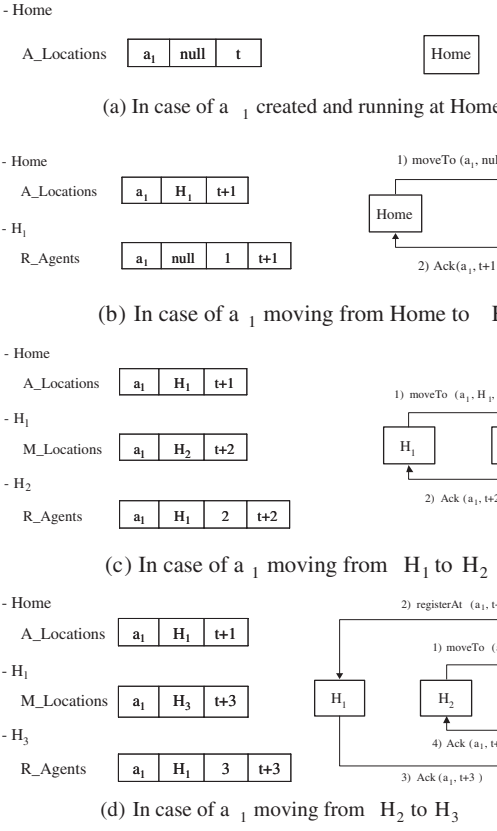


Fig. 1. An example of a_1 moving from $Home$ through H_3

service host where agent *aid* is running. *t_time* is a timestamp associated with agent *aid* when the agent is located at host H_{c_hid}.

• $N_ForwardP_i$: a set for saving forwarding pointer of every mobile agent whose forwarder H_i is. Its element is a tuple (*aid*, *f_hid*, *t_time*). *f_hid* is the identifier of the next forwarder having a forwarding pointer for agent *aid*. *t_time* is a timestamp associated with agent *aid* when the agent is running on host H_{f_hid}.

Let us illustrate how the two directory service protocols operate with the previously stated desirable features using figures 1 through 3. In these figures, we assume k is set to 3. Figure 1 shows an example that agent a_1 moves from its home to H_1 through H_3. In figure 1(a), agent a_1 is created and running at its home host, $Home$. Suppose its current timestamp is t. Then, in figure 1(b), agent a_1 moves from its home to H_1 by calling procedure MOVETO(). In this case, the timestamp becomes $(t + 1)$ and the current number of visiting hosts, *h_cnt*, is initialized to 1. After the agent migration finished, the home host knows agent a_1 is on H_1. In figure 1(c), agent a_1 moves from H_1 to H_2 in the same way.

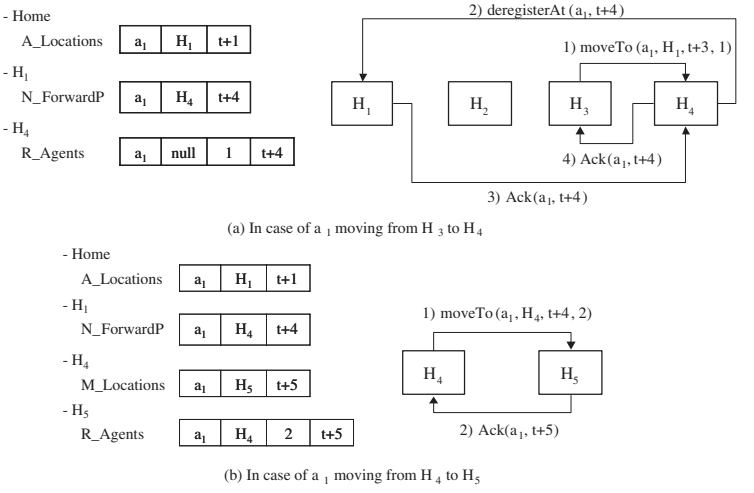
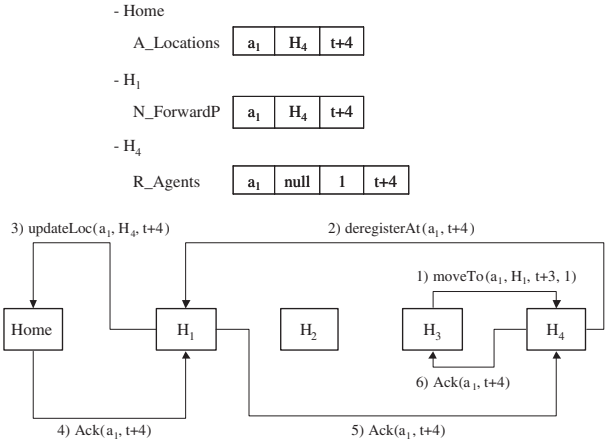


Fig. 2. An example of a_1 moving from H_3 through H_5 in $DSPNHU$

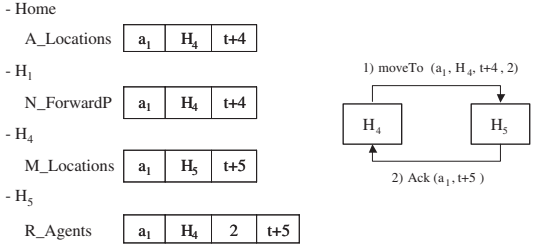
In this case, both the timestamp and *h_cnt* are all incremented by one. Also, H_2 is aware that H_1 is managing a_1's current location and then H_1 knows a_1 is on H_2. In figure 1(d), a_1 moves from H_2 to H_3, which is the secondly visiting host from H_1. In this case, the timestamp and *h_cnt* are also both incremented and then H_3 informs H_1 that agent a_1 with the timestamp is currently on H_3 by invoking procedure REGISTERAT(). After receiving the corresponding **Ack** message, H_3 forces H_2 to remove a_1's location information from its storage. Thus, the two proposed protocols allow only a_1's home host and H_1 to maintain a_1's forwarding pointer.

When every k-th agent migration is performed, the two protocols execute in different manners. Figures 2 and 3 illustrate their features respectively. Figure 2 shows agent a_1 migrates from H_3 to H_4 and H_5 in the protocol $DSPNHU$. In figure 2(a), a_1 moves from H_3 to H_4, the thirdly visiting host from H_1. In this case, as *h_cnt* in H_3 is the maximum number 3, H_3 resets the variable to 1 and informs H_4 of the fact through the procedure MOVETO(). This causes H_4 to notify H_1 that H_4 will be a_1's location manager from now by calling procedure DEREGISTERAT(). Then, forwarder H_1 saves a_1's forwarding pointer into the table $N_ForwardP$. In figure 2(b), a_1 moves from H_4 to H_5, the firstly visiting host from H_4. In this case, a_1's timestamp and *h_cnt* are both incremented and H_5 is aware that a_1's current location is managed by H_4.

However, in $DSPNHU$, a_1's home host knows that a_1 is currently on H_1. In this case, when another agent a_2 attempts to send its first message to a_1 to initially communicate with each other, the message must be eventually delivered to H_5 via a_1's home, H_1 and then H_4. If a_1 has migrated to a large number of service hosts, the initial message delivery time may significantly increase because the message should be delivered to a_1 via a certain number of forwarders(this case will be more concretely explained in section 3). To address this problem, the



(a) In case of a $_1$ moving from H_3 to H_4



(b) In case of a $_1$ moving from H_4 to H_5

Fig. 3. An example of a_1 moving from H_3 through H_5 in $DSPHU$

protocol $DSPHU$ is proposed. Figure 3 shows agent a_1 migrates from H_3 to H_4 and H_5 in the protocol. When a_1 moves from H_3 to H_4 in figure 3(a), the protocol performs the almost same procedure of $DSPNHU$ in figure 3(a). However, the only difference is that $DSPHU$ forces H_1 to inform a_1's home host of a_1's current location with its timestamp by invoking procedure UPDATELOC() like in figure 3(a). Therefore, $DSPHU$ enables each first message to be delivered to the target mobile agent via at most two additional hosts, a_1's home and location manager.

3 The Optimized Message Delivery Protocol

Our optimized message delivery protocol($OMDP$) executes basically based on the two proposed directory service protocols. Additionally, it enables each sending agent to communicate with mobile agents very fast by effectively using their bindings in the sending agent's location cache called C_Agents as follows.

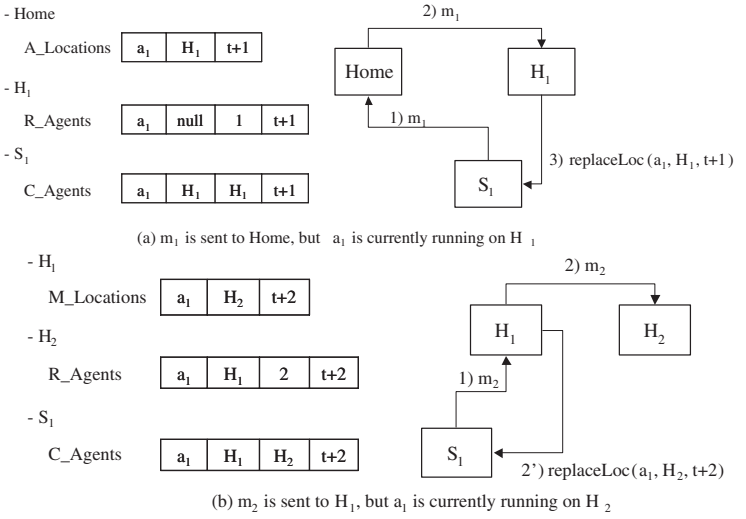
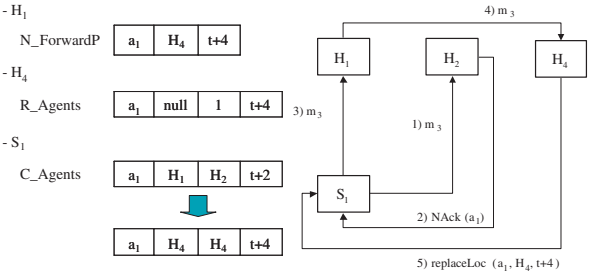


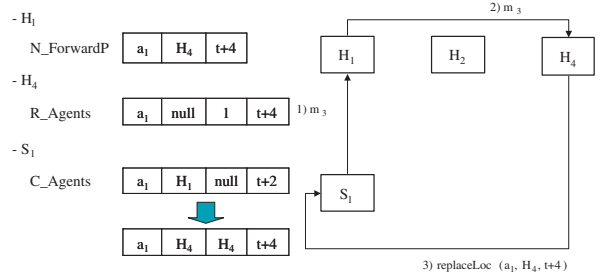
Fig. 4. In case of an agent at S_1 sending two messages m_1 and m_2 to agent a_1

• C_Agents_i: A table for saving location information of each mobile agent which agents running on H_i communicate with. Its element is a tuple (*aid*, *f_hid*, *c_hid*, *t_time*). The third field for agent *aid*, *c_hid*, is the identifier of the host where H_i knows the agent is currently located and running. Thus, each agent on H_i sends messages to *c_hid* to deliver them to agent *aid*. But, as this field is associated with a certain timeout value, it becomes invalid when its timer expires like soft state, and then is able to be used no longer. In this case, each agent on H_i uses the second field *f_hid*, the identifier of the forwarder of agent *aid*, in order to communicate with the agent. Thus, if it sends any message to *f_hid*, the message is eventually delivered to agent *aid*. *t_time* is a timestamp associated with agent *aid* when the agent is located at host H_{c_hid}.

We intend to clarify the effectiveness of our message delivery protocol $OMDP$ using figures 4 through 6. For example, figure 4 illustrates the basic operations of the protocol $OMDP$. In figure 4(a), an agent c at S_1 attempts to send message m_1 to a_1. Suppose that S_1 has no binding for a_1 in its location cache. In this case, agent c must send m_1 to a_1's home host. Then, m_1 is forwarded to H_1, where a_1 is running and which is currently a_1's location manager. At this point, H_1 informs S_1 of the identifiers of a_1's location manager(H_1) and current service host(H_1). In figure 4(b), agent c delivers message m_2 to a_1. In this case, c sends m_2 to H_1 because it knows a_1 is currently on H_1 like in figure 4(a). Then, H_1 forwards the message to H_2 by looking up a_1's pointer in the agent location table $M_Locations$. Simultaneously, location manager H_1 gives a_1's recent location information to S_1, which sets the current service host identifier of a_1 to H_2. Afterwards, agent c can directly communicate with a_1 at H_2 as long as a_1 is running on H_2.



(a) In case that a_1's current service host identifier is valid



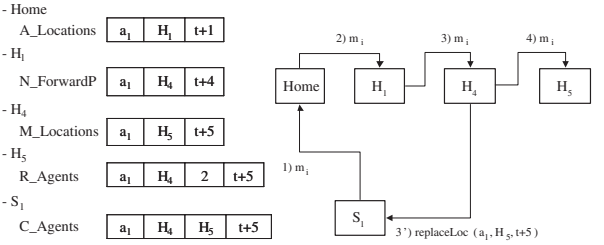
(b) In case that a_1's current service host identifier is invalid

Fig. 5. Two cases when agent c at S_1 sends message m_3 to agent a_1 by looking up a_1's binding in S_1's location cache

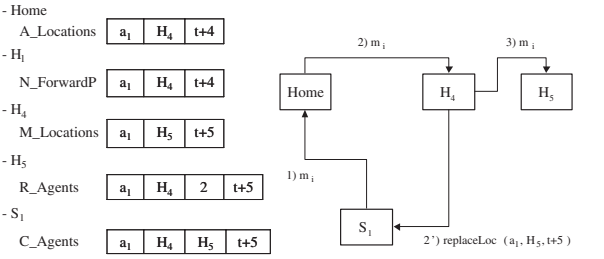
However, suppose that a_1 migrates from H_2 to H_4 via H_3 like in figure 5(a) without any further communication with agent c. Afterwards, if c sends message m_3 to H_2 after looking up a_1's current service host from S_1's location cache, it receives a negative **Ack** message from H_2 because a_1 is not on H_2 and H_2 has no location information for a_1. Thus, after finding a_1's forwarder H_1 from S_1's cache, c sends m_3 to H_1. Then, H_1 forwards the message to H_4 by a_1's forwarding pointer. Afterwards, H_4 notifies S_1 that H_4 is a_1's current service host and location manager.

If a_1's current host identifier has become invalid before c delivers m_3 to a_1 like in figure 5(b), c sends the message directly to H_1 based on a_1's forwarder lookup from S_1's location cache. Afterwards, S_1 eventually receives from H_4 the identifiers of a_1's current service host(H_4) and location manager(H_4) and updates the corresponding element in S_1's cache by using the recent information. This revalidation policy may reduce the rate where the communication failure case by out-of-date location information like in figure 5(a) occurs when agents is highly mobile.

Next, let us show how the message delivery protocol operates based on two different directory service protocols. In figure 6, agent c at S_1 sends message m_i to a_1 running on H_5 like in figures 2(b) and 3(b), but there is not a_1's binding in S_1's cache. After c has sent m_i to a_1's home host, the two directory service protocols allow m_i to be delivered to a_1 at H_5 in different ways. In the protocol



(a) In case of the protocol DSPNHU when m_i is sent to Home, but a_1 is currently running on H_5



(b) In case of the protocol DSPHU when m_i is sent to Home, but a_1 is currently running on H_5

Fig. 6. Examples showing the differences between two proposed directory service protocols in case of message delivery

$DSPNHU$, a_1's home knows a_1 is on H_1 like in figure 6(a). Thus, m_i is forwarded to H_1, which knows only the next forwarder of a_1, H_4. Then, H_1 sends m_i to H_4, a_1's current location manager. Thus, H_4 can forward the message to H_5 that a_1 is currently running on. Also, it informs S_1 of a_1's current location information. But, the protocol $DSPHU$ enables a_1's home to forward m_i directly to H_4 based on a_1's current location manager identifier like in figure 6(b).

4 Conclusion

In this paper, two directory service protocols $DSPNHU$ and $DSPHU$ and one message delivery protocol based on them, $OMDP$, were designed. $DSPNHU$ and $DSPHU$ considerably reduce both the length of chains of forwarding pointers by forcing the corresponding service host to maintain each mobile agent's pointer only after every k migrations of the agent have been finished. Thus, they can significantly reduce each message forwarding overhead and per-host storage maintenance cost of increasing chains of pointers. But, $DSPHU$ forces the previous location manager of each mobile agent to notify its home of the new location manager's identifier whereas $DSPNHU$ doesn't execute this update at the home. Thus, their respective behaviors cause the two protocols to have different tradeoffs with respect to home update overhead and initial communication

cost. Finally, $OMDP$ allows each sending agent to communicate with mobile agents very fast by effectively using their bindings in its location cache.

In this paper, the proposed protocols consider only agent mobility, not failures of directory service hosts. Thus, we are currently extending them to support the latter issue using some efficient redundancy techniques. Also, to evaluate their performance, we are implementing the protocols using a java-based mobile code toolkit, μCode[10].

References

1. J. Ahn. The Design of Efficient Directory Service and Message Delivery Protocols for Mobile Agents. *Technical Report KGU-CS-03-40*, Kyonggi University, 2003.
2. F. Baschieri, P. Bellavista and A. Corradi. Mobile Agents for Qos Tailoring, Control and Adaptation over the Internet: The UbiQoS Video on Demand Serbvice. *In Proc. of the 2nd International Symposium on Applications and the Internet*, pp. 109–118, 2002.
3. A. Fuggetta, G.P.Picco and G. Vigna. Understanding Code Mobility. *IEEE Transactions on Software Engineering*, Vol. 24, No. 5, pp. 342–361, 1998.
4. D. Lange and M. Oshima. *Programming and Deploying Mobile Agents with Aglets.* Addison-Wesley, 1998.
5. L. Moreau. Distributed Directory Service and Message Router for Mobile Agents. *Science of Computer Programming*, Vol. 39, No. 2-3, pp. 249–272, 2001.
6. L. Moreau. and D. Ribbens. Mobile Objects in Java. *Scientific Programming*, Vol. 10 , No. 1, pp. 91–100, 2002.
7. ObjectSpace. Voyager. *http://www.objectspace.com/.*
8. V. Pham and A. Karmouch. Mobile Software Agents: An Overview. *IEEE Communications Magazine*, Vol. 36, pp. 26–37, 1998.
9. G. P. Picco. Mobile Agents: An Introduction. *Journal of Microprocessors and Microsystems*, Vol.25, No.2, pp. 65–74, April 2001.
10. G. P. Picco. μCode: A Lightweight and Flexible Mobile Code Toolkit. *Lecture Notes In Computer Science*, Vol.1477, pp. 160–171, September 1998.
11. K. Rothermel and M. Schwehm. Mobile Agents. *Encyclopedia for Computer Science and Technology*, Vol. 40, pp. 155–176, 1999.
12. L.M. Silva, P. Simoes, G. Soares, P. Martins, V. Batista, C. Renato, L. Almeida, N. Stohr. JAMES: A Platform of Mobile Agents for the Management of Telecommunication Networks. *Lecture Notes In Computer Science*, 1699, 1999.

Optimization of Usability on an Authentication System Built from Voice and Neural Networks

Tae-Seung Lee and Byong-Won Hwang

School of Electronics, Telecommunication and Computer Engineering, Hankuk Aviation University, 200-1, Hwajeon-dong, Deokyang-gu, Koyang-city, Kyonggi-do, 412-791, Korea
thestaff@hitel.net, bwhwang@mail.hangkong.ac.kr

Abstract. While multilayer perceptrons (MLPs) have great possibility on the application to speaker verification, they suffer from an inferior learning speed. To appeal to users, the speaker verification systems based on MLPs must achieve a reasonable speed of user enrolling and it is thoroughly dependent on fast learning of MLPs. To attain real-time enrollment for the systems, the previous two studies, the discriminative cohort speakers (DCS) method and the omitting patterns in instant learning (OIL) method, have been devoted to the problem and each satisfied that objective. In this paper, we combine the two methods and apply the combination to the systems, assuming that the two methods operate on different optimization principles. Through experiment on real speech database using an MLP-based speaker verification system to which the combination is applied, the feasibility of the combination is verified from the results.

Keywords: Biometric authentication system, speaker verification, multiplayer perceptrons, error backpropagation, real-time enrollment, discriminative cohort speakers, omitting patterns in instant learning

1 Introduction

Speaker verification systems require real-time speaker enrollment or adaptation as well as real-time verification to provide a satisfactory usability. To be used in daily life, it is necessary that speaker verification systems have not only fast verifications but also short enrollments of speakers. Most users want to access secured facility just after enrolling themselves for the system. If they have to wait for a long time for the first access, they may quit their enrolling process. Moreover, the voice of the same speaker can be changed due to senility, disease or any time-relevant factors. To adapt for such variability, many speaker verification algorithms have introduced adaptation methods which use the recent voices of the enrolled speakers to update their vocal characteristics [1], [2], [3]. In the situation, the fast enrollment becomes more serious because adaptation can be considered as refinement of the earlier enrollment.

Unlike parametric-based speaker verification systems, the systems based on multilayer perceptrons (MLPs) more quickly conduct the computation needed to verify identities but slowly to enroll speakers [4], [5]. An MLP consists of one input layer, more than zero hidden layer(s) and one output layer. Input layer receives pattern signal, the hidden layer determines the network's behavior and the output

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 386–395, 2004.
© Springer-Verlag Berlin Heidelberg 2004

layer presents final decisions for the input. Each layer consists of more than one computational node and all nodes in a layer are fully connected with the nodes of the facing layers. This structure makes output nodes share the computational nodes in input and hidden layers with each other, so inspires a fast verification process even with low-computational capability. On the other hand, it is difficult to settle optimal values of internally weighted connections between nodes to achieve the best decisions of all output nodes. In addition, a large number of background speakers required for an MLP to learn an enrolling speaker makes the difficulty get worse in MLP-based speaker verification systems.

To solve the difficulty in settling optimal values of internally weighted connections, Lee et al. attempted to reduce the number of learning steps and shorten the duration of each learning step in the online mode error backpropagation (EBP) algorithm [6]. The EBP algorithm is widely used to train MLPs but has a poor learning speed due to its dependency on local gradient information. Nevertheless, the EBP has advantages to drive an excellent anti-overfitting ability and reveals a fairly fast learning when it is operated in online mode on applications of pattern recognition [7], [8]. The one of two method Lee et al. proposed for improving the EBP algorithm, called the omitting patterns in instant learning (OIL) method, is to exploit the redundancy of pattern recognition data and achieved a substantial improvement in learning speed without losing any recognition rate.

To relieve the awful amount of background speakers that hinders speakers from enrolling in real-time for MLP-based speaker verification systems, Lee et al. sought to reduce the number of background speakers required to enroll speakers and the attempt proved to be successful [9], [10]. MLPs learn an enrolling speaker by the difference to any other speakers, therefore background speakers should be provided sufficiently to represent speakers of the whole world. However, the increasing number of background speakers means increasing of learning data and it is not acceptable for MLP-based speaker verification systems that must enroll speakers in real-time. The data reduction method Lee et al. introduced for relieving the burden, called the discriminative cohort speakers (DCS) method, is to select the very background speakers related to the enrolling speaker in order to make use of the discriminant learning property of MLPs and obtained a rather effective result in enrolling speed.

In this paper we combine the two methods into a hybrid method to get further improvement in enrolling speed on MLP-based speaker verification, assuming that the two methods operate on different optimization principles. The DCS to select background speakers on a qualitative criterion cuts off irrelevant learning data before actual learning starts; hence it is considered a global optimization for learning data [10]. Then the learning begins and useless learning data out of the learning data which is globally optimized once and makes up one learning epoch in the EBP are omitted from each learning step by the OIL, therefore it is considered a local optimization for learning data [6]. When the two methods are combined, the optimality in data amount involved in MLP learning can be maximized and the higher performance for real-time enrollment can be easy to reach. To evaluate the improvement of the combination, experiment is designed which compare the performance of the combination with those of the individual methods using an implemented MLP-based speaker verification system and a Korean speech database.

This paper hereafter is organized as follows. In Sections 2 and 3 we introduce the reduction method of learning data and the omitting method of useless data, respectively. Then the implemented MLP-based speaker verification system to which

the two methods are applied is described in Section 4. Using the system, an experiment is conducted to verify the reduction effect in enrolling duration by combining the methods in Section 5. The paper is finally summarized in Section 6.

2 Discriminative Cohort Speakers Method

The prospect to reduce background speakers in MLP-based speaker verifications arises from the geometric contiguity of learning models. That is, in MLP learning, the learning of a model is cooperated only with its geometrically contiguous models. When an enrolling speaker is given into background speaker crowd for its learning, the decision boundary of an MLP to learn the difference between the enrolling speaker and background speakers is affected only by the background speakers adjacent to the enrolling speaker. If a great number of background speakers are reserved in the system to obtain very low verification error, the percentage of such background speakers does increase and the number of background speakers needed to establish final decision boundary can be shortened.

The process of the DCS to select the background speakers similar to an enrolling speaker in MLP-based speaker verifications is implemented like this:

$$\begin{aligned} S_{Cohort} &= Sel_{M_{MLP} \geq \theta, I}(Sort_{Dec}(M_{MLP}(S_{BG} \mid \mathbf{X}))), \\ S_{BG} &= \{S_i \mid 1 \leq i \leq I\} \end{aligned} \quad . \tag{1}$$

where, $\mathbf{X}$ is the speech of enrolling speaker, S_{BG} the background speakers set which population is I, M_{MLP} the MLP function which evaluates likelihoods of the given $\mathbf{X}$ to the background speakers. $Sort_{Dec}$ stands for the function to sort given values in descending manner, $Sel_{M_{MLP} \geq \theta, I}$ for the function to select relevant background speakers whose M_{MLP}s exceed the preset threshold θ.

In this paper, MLPs to calculate M_{MLP} are called MLP-I and MLPs to learn an enrolling speaker using the background speakers selected by MLP-I called MLP-II. While MLP-Is are learned before enrolling speakers using background speakers' data, MLP-IIs are learned at enrolling speakers. It should be noted that although an MLP-II has one output node since it discriminates the current pattern input just into the enrolled speaker model and the background speaker model, an MLP-I has I output nodes since it has to evaluate likelihoods of given speech pattern to all background speakers.

3 Omitting Patterns in Instant Learning Method

MLPs learn the representation of models by establishing decision boundary which discriminates geometrically the model areas. If patterns of all models are fully presented in iterative manner and the internal learnable weights of an MLP are adjusted so that all the patterns of each model are classified into its own model area, the decision boundary can be finally settled in an optimal position.

The online mode EBP algorithm, the common method for MLP learning, updates the weights of an MLP using the information related to the given pattern and the current weights status like this:

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) + \Delta w_{ij}(n) \\ &= w_{ij}(n) - \eta \frac{\partial e_p(n)}{\partial w_{ij}(n)} \cdot \end{aligned} \tag{2}$$

$$e_p(n) = \frac{1}{2}\sum_{k=1}^{M} e_k^2(n) \cdot \tag{3}$$

$$e_k(n) = d_k(n) - y_k(n) \cdot \tag{4}$$

where, w_{ij} is the weight to link with a weighted value from computational node j to node i, n the update count of weights, and e_p the summed error from all output nodes for the given pattern p. e_k, d_k and y_k are the error, the learning objective and the network value of the output node k, respectively. M designates the number of output nodes and η the learning rate to determine how much portion of the weight vector change Δw_{ij} is applied to the update. The learning objective is, in general, designated 1 if the output node corresponds to the model of the current pattern, otherwise 0 or -1 according to the type of activation function: binary type or bipolar type. Weight updates continue until some criterions are satisfied, for example, the summation of e_ps for all learning patterns goes down below a certain value. After a learning is complete, network outputs are converging to their own objectives, which are derived from the learned weights, and decision boundary is formed at the valley between the highest output values on each model area.

The usefulness of the given pattern in the current epoch can be determined on the criterion of error energy objective. One epoch is defined as the duration that all learning patterns are once presented and the evaluation of when the learning stops is carried out on the end of each epoch. In the online mode EBP, the achievement of learning in the current epoch is measured with the error energy averaged for the entire N patterns like this:

$$\begin{aligned} e_{avg}(t) &= \frac{1}{N}\sum_{p=1}^{N} e_p(t) \\ &= \frac{1}{2N}\sum_{p=1}^{N}\sum_{k=1}^{M} e_k^2(t) \end{aligned} \quad . \tag{5}$$

where, t is the epoch count. Learning continues until the average error energy $e_{avg}(t)$ is less than the learning objective e_{obj}:

$$\begin{cases} w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n), & if\ e_{avg}(t) > e_{obj} \\ Stops, & otherwise \end{cases} . \tag{6}$$

The relationship between the average error energy and the individual error energies can be described like this:

$$\begin{aligned} & e_{avg}(t) \le e_{obj}, \\ & if\ e_C^2(n) \le 2\lambda e_{obj}\ \ for\ all\ N\ \ patterns, \\ & 0 < \lambda \le 1, \end{aligned} \tag{7}$$

where, $e_C^2(n)$ is the error energy of the output node C associated with given pattern. This expression means that if the $e_C^2(n)$s for all learning patterns are less than or equal to $2e_{obj}$, then the learning is complete, assuming that the learning is progressed sufficiently to ignore the other output values beside C. As a result, it is possible to learn only the patterns with $e_C^2(n) > 2e_{obj}$ to complete the learning. In the equation the coefficient λ is inserted to determine the depth of patterns which weight vector is to be updated. When λ is near 1, the number of omitted patterns increases but the count of learning epochs increases as well. Hence it is necessary to search for a proper λ to achieve the minimum count of learning epochs and the maximum number of omitted patterns so that the shortest learning duration is obtained. MLP-IIs described in Section 2 are trained by this method.

4 Implemented System

The speaker verification system extracts isolated words from input utterances, classifies the isolated words into nine Korean continuants (/a/, /e/, /ə/, /o/, /u/, /ɨ/, /i/, /l/, nasals) stream, learns an enrolling speaker using MLPs for each continuant, and calculates identity scores of customers. The procedure performed in this system is outlined in Fig. 1 and each process is described in the following:

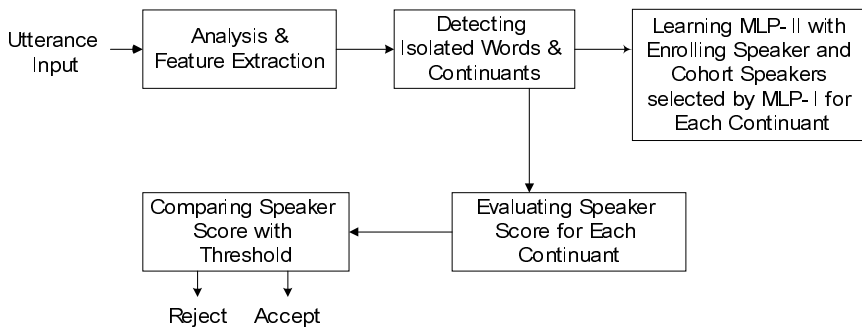


Fig. 1. The process flow of the MLP-based speaker verification system

(1) Analysis and Feature Extraction [11]
The utterance input sampled in 16 bits and 16 kHz is divided into 30 ms frames overlapped every 10 ms. 16 Mel-scaled filter bank coefficients are extracted from each frame and are used to detect isolated words and continuants. To remove the effect of utterance loudness from the entire spectrum envelope, the average of the coefficients from 0 to 1 kHz is subtracted from all the coefficients and the coefficients are adjusted for the average of the whole coefficients to be zero. 50 Mel-scaled filter

bank coefficients that are especially linear scaled from 0 to 3 kHz are extracted from each frame and are used for speaker verification. This scaling adopts the study arguing that more information about speakers concentrates on the second formant rather than the first [12]. As with the extraction to detect isolated words and continuants, the same process to remove the effect of utterance loudness is applied here too.

(2) Detecting Isolated Words and Continuants
Isolated words and continuants are detected using an MLP learned to detect all the continuants and silence in speaker-independent mode.

(3) Learning MLP-II with Enrolling Speaker for Each Continuant
For each continuant, the continuants detected from the isolated words are input to corresponding MLP-I and outputs of the MLP-I are averaged. Then the background speakers to present their output averages more than the preset threshold θ are selected. MLP-IIs learn enrolling speaker with the selected background speakers for each continuant.

(4) Evaluating Speaker Score for Each Continuant
For each continuant, the all the frames detected from the isolated words are input to the corresponding MLP-II. All the outputs of the MLPs are averaged.

(5) Comparing Speaker Score with Threshold
The final reject/accept decision is made by comparing a predefined threshold with the average of the step (4).

Since this speaker verification system uses the continuants as speaker recognition units, the underlying densities show mono-modal distributions [13]. It is, therefore, enough for each MLP to have two layers structure that includes one hidden layer [14], [15]. Since the number of models for the MLPs to learn is two: one is enrolling speaker and the other background speakers, the MLPs can learn the models by only one output node and two hidden nodes. Nine MLPs in total are provided for nine continuants.

5 Experiment

In the experiment, the improvement of enrolling duration by the combination along with the improvements of individual methods are measured and compared with the online EBP algorithm. To evaluate the improvements, experiment is designed for the implemented system and a Korean speech database. This section records the results of the evaluation.

5.1 Speech Database

The speech data used in this experiment are the recorded voice of connected four digits, spoken by 40 Korean male and female speakers. The digits are ten Arabic

numerals pronounced in Korean as /goN/, /il/, /i/, /sam/, /sa/, /o/, /yug/, /cil/, /pal/, /gu/, each corresponding to a digit from 0 to 9. Each speaker utters 35 words of different 4-digit strings four times, when the utterance is recorded in 16 bits resolution and 16 kHz sampling. Three of the four utterance samples are used to enroll the speaker, and the last utterance is used for verification. In order to learn the enrolling speakers discriminatively, additional 29 male and female speakers are participated as background speakers for MLPs other than the above 40 speakers.

5.2 Experiment Conditions

In our experiment, the conditions for learning MLPs to enroll a speaker are set up as follows [5]:

- Input patterns are normalized such that the elements of each pattern vector are placed into the range from –1.0 to +1.0.
- The learning targets of output node are +0.9 for the enrolling speaker and –0.9 for the background speakers to obtain faster speed in EBP learning.
- Speech patterns are presented in an alternating fashion for the two models during learning. In most cases, however, the numbers of patterns for the two models are not the same. Accordingly, the patterns are presented repetitively (more than once) for the model with fewer patterns until all the patterns have been presented once for the model with more patterns. This completes one epoch of learning.
- Since learning may stop at a local minimum, the epochs of learning are limited to 1000 maximum.

In our experiment, each of the 40 speakers can be treated as both the enrolling speaker and the test speaker. When one of them is picked as the test speaker, then the other 39 speakers are used as imposters. As a result, 35 tests using the 35 words are performed for a true speaker and 1,365 (35 * 39) tests for the imposters. In total, we performed 1,400 (35 * 40) trials of test for true speaker and 54,600 (35 * 40 * 39) trials for imposters.

The experiment is conducted on a 1 GHz personal computer machine. In the experiment result, the error rate designates equal error rate, the number of learning epochs the averaged number of epochs used to enroll a speaker for a digit string, the number of learning patterns the averaged number of patterns for the same string, and the learning duration the overall duration taken to learn those patterns. The values of error rate, the number of learning epochs, the number of learning patterns, and learning durations are the averages for the results of three-time learning each with the same MLP learning condition to compensate for the effect of the randomly selected initial weights.

5.3 Results

Experiments are conducted to evaluate the performances of the online EBP, the OIL, and the DCS combined with the OIL. The results of all experiments are presented in Fig. 2. In the figure, OnEBP designates the online EBP, the numbers of the bottom the preset thresholds in the DCS. The performance of the online EBP is evaluated with

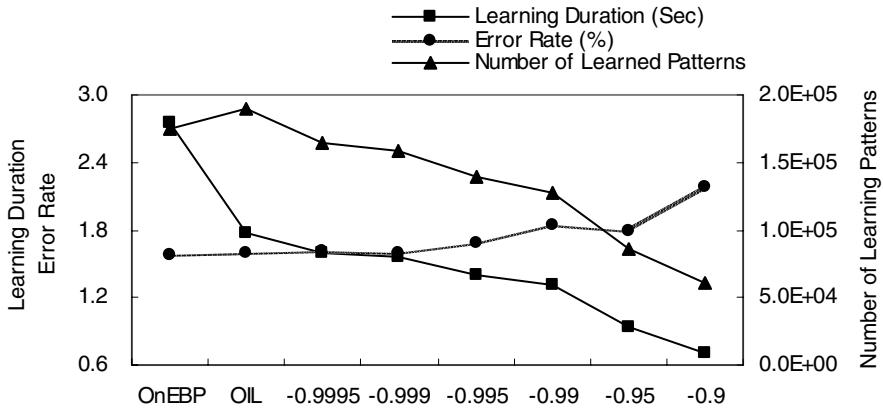


Fig. 2. Experimental results of the online EBP, the OIL, and the DCS with the OIL

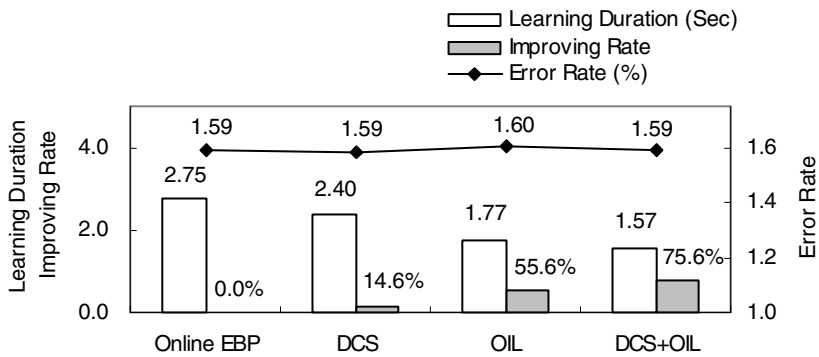


Fig. 3. Performance comparison of all methods

the optimized learning parameters, i.e. learning rate is 0.5 and learning objective error energy 0.005 as searched in [16]. The figures for the OIL performance are measured with $\lambda = 0.3$ for learning rate 1 and learning objective error energy 0.005. In the measurements of the DCS combined with the OIL, the optimal result can be taken at the threshold –0.999 because the numbers over this point make higher verification errors. On the basis of the online EBP algorithm, the OIL achieves a quite improvement in enrolling duration without making verification error worse. With the OIL applied, the DCS keeps the learning duration decreasing as the threshold increases. From the results, it can be known that the combination of the two methods is effective to shorten the enrolling duration over the individual methods.

The performance evaluations are summarized in Fig. 3. With the same level of verification error as the online EBP, the DCS marks the improvement of 14.6 % and the OIL 55.6 % over the online EBP. The combination of the two methods further improves enrolling duration by 75.6% over the online EBP. The better result of the combination to those of the OIL and the DCS demonstrates that the two methods operate on different optimization principles and make a synergy when they are employed together.

6 Conclusion

So far real-time speaker enrolling problem has been attempted to provide higher usability for MLP-based speaker verification systems. While MLPs have great potential on the application to speaker verification, they suffer from poor learning speed. Many users may call for an instant enrolling for speaker verification system, hence it is necessary for the defect of MLPs to be reformed. To solve the problem, we fused the existing two methods, the DCS and the OIL, to enhance the speaker enrolling speed for MLP-based speaker verification systems. From the results of experiment on the real speech database, it was acquired that the previous methods were based on distinct reduction bases and it can be concluded that the combination of the methods is more effective to shorten speaker enrolling duration for the speaker verification systems based on MLPs.

References

1. Matsui, T., Aikawa, K.: Robust Model for Speaker Verification against Session-Dependent Utterance Variation. IEEE International Conference on Acoustics, Speech and Signal Processing 1 (1998) 117–120
2. Mistretta, W., Farrell, K.: Model Adaptation Methods for Speaker Verification. IEEE International Conference on Acoustics, Speech and Signal Processing 1 (1998) 113–116
3. Matsui, T., Furui, S.: Speaker Adaptation of Tied-Mixture-Based Phoneme Models for Text-Prompted Speaker Recognition. IEEE International Conference on Acoustics, Speech and Signal Processing 1 (1994) 125–128
4. Rosenberg, A. E., Parthasarathy, S.: Speaker Background Models for Connected Digit Password Speaker Verification. IEEE International Conference on Acoustics, Speech, and Signal Processing 1 (1996) 81–84
5. Bengio, Y.: Neural Networks for Speech and Sequence Recognition. International Thomson Computer Press, London Boston (1995)
6. Lee, T., Choi, H., Kwag, Y., Hwang, B.: A Method on Improvement of the Online Mode Error Backpropagation Algorithm for Pattern Recognition. Lecture Notes in Artificial Intelligence 2417 (2002) 275–284
7. Lawrence, S., Giles, C. L.: Overfitting and Neural Networks: Conjugate Gradient and Backpropagation. IEEE-INNS-ENNS International Joint Conference on Neural Networks 1 (2000) 114–119
8. LeCun, Y.: Generalization and Network Design Strategies. Department of Computer Science, University of Toronto (1989)
9. Lee, T., Choi, S., Choi, W., Park, H., Lim, S., Hwang, B.: Faster Speaker Enrollment for Speaker Verification Systems Based on MLPs by Using Discriminative Cohort Speakers Method. Lecture Notes in Artificial Intelligence 2718 (2003) 734–743
10. Lee, T., Choi, S., Choi, W., Park, H., Lim, S., Hwang, B.: A Qualitative Discriminative Cohort Speakers Method to Reduce Learning Data for MLP-Based Speaker Verification Systems. Lecture Notes in Computer Science 2690 (2003) 1082–1086
11. Becchetti, C., Ricotti, L. P.: Speech Recognition: Theory and C++ Implementation. John Wiley & Sons, Chinchester New York Weinheim Brisbane Singapore Toronto (1999)
12. Cristea, P., Valsan, Z.: New Cepstrum Frequency Scale for Neural Network Speaker Verification. IEEE International Conference on Electronics, Circuits and Systems 3 (1999) 1573–1576
13. Savic, M., Sorensen, J.: Phoneme Based Speaker Verification. IEEE International Conference on Acoustics, Speech, and Signal Processing 2 (1992) 165–168

14. Delacretaz, D. P., Hennebert, J.: Text-Prompted Speaker Verification Experiments with Phoneme Specific MLPs. IEEE International Conference on Acoustics, Speech, and Signal Processing 2 (1998) 777–780
15. Lippmann, R. P.: An Introduction to Computing with Neural Nets. IEEE Acoustics, Speech, and Signal Processing Magazine 4 (1987) 4–22
16. Lee, T., Hwang, B.: Continuants Based Neural Speaker Verification System. To be published in Lecture Notes in Artificial Intelligence (2004)

An Efficient Simple Cooling Schedule for Simulated Annealing

Mir M. Atiqullah

Aerospace and Mechanical Engineering Department
Parks College of Engineering and Aviation
Saint Louis University
Saint Louis, MO 63103
atiqulla@slu.edu

Abstract. The capability of Global solution of an optimization problem is the forte of Simulated Annealing (SA). Theoretically only the infinite-time algorithm can guarantee the global solution. The finite-time characteristics of the algorithm depend largely on the ensemble of certain control parameters. Since the main parameter is dubbed temperature, the dynamics of how it is manipulated is widely known as cooling schedule. A variety of methods, from simple geometric to highly complex, have been proposed in the literature. While global solution capability has been the overall goal for all implementation, few schedules combined effective solution with simplicity of the cooling schedule. A novel schedule is proposed which combines efficiency with simplicity into an easily implementable algorithm. Several fundamental cooling schemes are compared with the proposed one based on 2 test problems. Our schedule faired competitively with most while being the simplest.

Keywords: Optimization, simulated annealing, cooling schedule.

1 Introduction

A cooling schedule is defined by a set of parameters governing the finite time behavior of the SA algorithm. The parameters imitate the asymptotic behavior of the homogeneous annealing algorithm using an inhomogeneous implementation. Although, the global converge guarantee is lost the finite time implementation gives results, which are at or very close to the global optimum. For engineering designs with multimodal objectives, this capability sets SA above most traditional optimization methods.

A finite time implementation of SA would consist of a series of homogeneous Markov chains where at each stage, transitions are generated according to the Metropolis criteria [1]. A finite time cooling schedule is constructed on the concept of quasi-equilibrium of the Markov chain, such that the probability distribution of the configuration is arbitrarily *close* to the stationary distribution of the objective function at a specific temperature. It is this *closeness* that has been used as the metric for development of various cooling schedules. The structure of cooling schedules for SA, implementing adaptive strategies, can be generalized by three factors:

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 396–404, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Temperature: Initially, a quasi-equilibrium state can be assumed by choosing the initial value of the temperature (control parameter) high enough such that virtually all transitions are accepted according to the Metropolis criteria. In this situation, all configurations exist with equal probability of acceptance. To simulate the convergence conditions of the homogeneous algorithm, the temperature must approach a value of zero as the algorithm progresses. In practice, the temperature value is reduced to sufficiently small values such that virtually no worse configurations are accepted by the Metropolis acceptance test and no further significant improvement of objective is expected.

Markov chain: The number of transitions attempted at any specific temperature is the length L_k of the *k*-th Markov chain. The chain length is governed by the notion of *closeness* of the current probability distribution a_{L_k,t_k} to the stationary distribution q_{t_k}. Various adaptive schedules have taken different approaches with different assumptions and preset conditions to determine when such *closeness* is achieved.

Temperature Decrement: This rule is also intimately related to the notion of quasi-equilibrium of the probability distribution of configurations. A large decrement of the temperature t_k at the *k*-th Markov chain will necessitate a large number of transitions before a quasi-equilibrium state would be restored. Most of the schedules adopt the strategy of small decrements in temperature t_k to avoid long Markov chains L_k. It is a trade-off between the temperature decrement rate and length of Markov chains.

2 A Parametric Cooling Schedule

Adaptive schedules generally probe the solution domain and set the cooling parameters. These methods use statistics and a series of approximations, which make the cooling schedule fairly complex [2]. Research is also reported on 'tuning' the annealing algorithm [3] which includes the cooling schedule. However, the performance of these schedules could not be distinguished significantly from ones using less rigorous mathematics. Using the simplicity of the static (non-adaptive) schedules and problem specific guidance from the adaptive schedules, we introduce a parametric cooling schedule.

2.1 Initial Temperature

The initial temperature t_0 should be high enough such that a quasi-equilibrium state can be claimed, and all configurations would be equally acceptable. In our approach, we include all cost increasing and cost decreasing moves to estimate the t_0. Therefore the expected change in the cost function, considering all transitions, can be computed as

$$< \Delta C > \approx \overline{\Delta C} = i, j \subset \Re : \frac{\sum \left| \Delta C_{i,j} \right|}{ntrials} \tag{1}$$

Her $\left|\Delta C_{i,j}\right|$ is the absolute value of the difference in cost value during transition from *i*th to *j*th configuration and *ntrials* is the number of times the various neighborhood configurations. Assuming that the configurations are normally distributed, and if $\sigma_{\Delta C}$ is the sample standard deviation of all the $\left|\Delta C\right|$ encountered, the initial acceptance ratio χ_0 can be computed by

$$\chi_0 = \frac{\text{no. of accepted moves}}{\text{no. of proposed moves}} \approx 1.00 \approx \exp\left\{-\frac{\left(\overline{\Delta C} + 3\sigma_{\Delta C}\right)}{t_0}\right\}, \tag{2}$$

which leads to the new rule for the computation of the initial temperature:

$$t_0 = \frac{\left(\overline{\Delta C} + 3\sigma_{\Delta C}\right)}{\ln\left(\frac{1}{\chi_0}\right)} \tag{3}$$

Experiments have suggested that the value of t_0 as calculated by above formula is about 50% higher than other adaptive schedules.

2.2 Length of the Markov Chain

The effort to arrive at quasi-equilibrium in a Markov chain is related to the size of neighborhood ($\Re$). We propose a Markov chain that can be terminated if either the number of acceptances or rejections reach a certain number $\Lambda\left|\Re\right|$, That is,

$$L_k = \begin{cases} m1 + \Lambda\left|\Re\right|; \ \left\{m2 = \Lambda\left|\Re\right|, \ m1 < m2\right\} \\ m2 + \Lambda\left|\Re\right|; \ \left\{m1 = \Lambda\left|\Re\right|, \ m2 < m1\right\} \end{cases} \tag{4}$$

where m1 and m2 are the cost decreasing and cost increasing moves. The value of multiplication factor Λ could be as small as 1, if decrement is executed carefully.

2.3 Decrement Rule

An annealing algorithm progresses in three distinct stages; global positioning, local search and refine solution. A cooling strategy should reflect these stages.

Any rapid decrement of the temperature would result in a 'quenching' effect and entrapment of the configuration locally. In the third stage, the temperature decrement rates should be maintained at lower values to result in a flat convergence pattern.

During the middle part of annealing, the algorithm should perform most of the necessary decrements in temperature and settle in the locality of the optimum. During annealing, the cost function is assumed to follow a Gaussian pattern notably at higher

temperatures. Hence a Gaussian-like temperature decrement rule is proposed over the entire annealing process. The following formula is proposed to compute the annealing temperature t_k during a Markov chain k:

$$t_k = t_0 \cdot a^{-\left[\frac{k}{f \cdot k\max}\right]^b} \tag{5}$$

where a and f are the control parameters and $kmax$ is the maximum number of chains to be executed. At the last chain, $t_k = t_f$ and $k = k_{max}$. Equations (5) yields,

$$t_k = t_0 \cdot a^{-\left[\frac{1}{f}\right]^b} \tag{6}$$

By rearranging and taking logarithms, we compute b as,

$$b = \frac{P}{Q} \tag{7}$$

where, $P = \ln\left\{\dfrac{\ln\left(\frac{t_0}{t_f}\right)}{\ln(a)}\right\}$ and $Q = \ln\left(\dfrac{1}{f}\right)$.

The only difficulty lies in the selection of the values of a and f. When the algorithm is in the kth Markov chain and the parameter f equals (k/k_{max}) and using b=1, the corresponding temperature is obtained by $t_k = t_0/a$. This indicates that the temperature attained in the kth chain is equal to the (1/a)th fraction of the initial temperature t_0. The point identified by (t_0/a) and $(f * k_{max})$, is the parametric control point. By shifting this point on the decrement plot, the overall cooling pattern can be manipulated. If one wishes to dwell less time at higher temperatures, the control point should be shifted in the general direction of the origin. This implies that either f is to be decreased or a is to be increased or both. For a typical control point, the temperature will be reduced to half of the initial temperature at about one-third the maximum number of allowed Markov chains, i.e., a = 2 and f = ⅓. One could use other sets for trials.

2.4 Stopping Criteria or Final Value of Temperature

Using a predetermined small value for the final temperature with a parametric decrement rule, an upper limit is chosen for the number of Markov chains. As such, the algorithm is terminated if any of the following criteria are met in the order listed below.

(a) The cost value in several (e.g. 5) consecutive Markov chains did not improve.
(b) Five consecutive Markov chains did not improve the average cost beyond a specified small fraction ε i.e.,

$$\frac{\overline{C}_{k-1} - \overline{C}_k}{\overline{C}_{k-1}} < \varepsilon \tag{8}$$

The value of ε is set from past experience based on the cost values, scale factors, the accuracy desired as well as the computational effort involved.

(c) The algorithm did not terminate in *kmax* Markov chains using the rules (a) and (b) above. If proper stopping criteria are used, it is unusual to have the algorithm stopped by this method, which may be an indication of insufficient annealing.

3 Performance of the Schedules

Performance of cooling schedules is subjective as there are multiple issues to be considered. Some schedules are based on more rigorous theory than others and one would generally tend to put more faith in them. The overall issue of the quality of a schedule is also linked with the problems used to establish the comparison. In this work, a study is performed to investigate the comparative effectiveness of several schedules by solving two combinatorial/discrete optimization problems. The first example relates to manufacturing and the second to a structural design.

3.1 Example 1: Optimization of Part Placement

A group of 8 cylindrical parts are to be machined at a time on a CNC machining center attached onto pallets on square grids 1 inch apart. Moreover there can not be any interference or overlap between the loaded parts. The objective is to determine the positions of the 8 parts of various sizes (i.e. diameters) on the pallets, such that the pallet with the minimum size (area) can be used.

For the simulated annealing to work, the initial (x,y) positions of the parts are needed. An arbitrary set of initial positions was randomly generated and shown in Table 1.

Table 1. Cylindrical part sizes and their initial positions

Part #	Part diameter (in)	Initial position of parts (in)		Objective value -Area (in^2)
		x	y	
1	3.00	2	4	1571.625
2	6.50	22	10	
3	7.25	38	23	
4	4.5	46	32	
5	5.0	10	30	
6	6.0	6	24	
7	9.5	22	15	
8	10.0	45	12	

The same initial position set is used in all test runs. There are 2x8 design variables (8 sets of x and y values) and *n(n-1)/2* interference or overlap constraints. Additionally there are 2 sets of constraints, preventing parts from going overboard. There are 16^{45} ($\approx$ 1.532E54) different possible combinations of the variables among which the optimum set must be found.

3.2 Example 2: Design of a Space Truss

This problem calls for weight minimization of a space truss [4]. There are 25 members in the truss divided into 8 groups of same size members which can assume sizes A_i from a finite discrete set of choices thus posing it as a discrete (combinatorial) optimization problem. The grouping of the truss members is organized as follows:

Group 1: A_1 Group 2: $A_2 = A_3 = A_4 = A_5$ Group 3: $A_6 = A_7 = A_8 = A_9$
Group 4: $A_{10} = A_{11}$ Group 5: $A_{12} = A_{13}$ Group 6: $A_{14} = A_{15} = A_{16} = A_{17}$
Group 7: $A_{18} = A_{19} = A_{20} = A_{21}$ Group 8: $A_{22} = A_{23} = A_{24} = A_{25}$

The allowable cross-sectional areas for each of the groups are assumed to vary from 0.1 to 2.6 in^2 in steps of 0.1 in^2, 2.8, 3.0, 3.2, and 3.4 in^2. Since each of the 8 design variables can assume any of the 30 allowable sizes, the variables may be combined in 30^8 ($\approx$ 6.56E11) possible configurations. The loads acting on the structure are shown in Table 2. The maximum allowable stress in the members is 40 ksi in both tension and compression. The top nodes (1 and 2) may not be displaced more than $\pm 0.35\ in.$ in either x or y direction. The Young's modulus of the member materials (E) is 10^7 psi and the weight density (ρ) is 0.1 lb./in^3. The objective is to minimize the weight of the truss, $\sum_{i=1}^{25} A_i l_i \rho$, where l_i is the length of the *i*th member.

Table 2. Loading on the 25-bar Truss

Node numbers	Loads along the axes (lbs)		
	X	Y	Z
1	1000	10,000	–10,000
2	0	10,000	–10,000
3	500	0	0
6	600	0	0

Several published cooling schedules along with some variants are implemented and are briefly described for easy reference as follows:

Schedule # 1: This schedule is implemented as proposed by Huang *et al*. [5]. Using values of $\lambda = 0.7$, as suggested by Huang, and $\lambda = 0.25$ for a tightened expected deviation.

Schedule # 2: This schedule is implemented as per the arguments made by Otten and van Ginneken [6]. The fixed length Markov chains are executed with a careful stepping of temperature while maintaining some quasi-equilibrium state. A control parameter *delta*, similar to the one used in schedule #4 , is used and experiments have been conducted using *delta* values of 1 and 5.

Schedule # 3: Proposed by Romeo and Sangiovanni-Vincentelli [7]. They used a value of t_0 that is computed in the current implementation by sampling the neighborhood objective values (costs). The value of t_0 made sure that at least 90% of the visited configurations are accepted at the initial temperature. Two values of the temperature decrement factor (0.9, and 0.95) are used.

Schedule # 4: This is same as the parametric schedule described earlier. The Markov chains are executed with the bounds $n\text{var} \leq L_k \leq 2 \cdot n\text{var}$, where *nvar* is the number of variables. In the case when the number of rejected or accepted moves equals *nvar* the chain is terminated, and none of these numbers can exceed twice the number of variables.

4 Results and Discussion

The results of execution of the simulated annealing algorithm using the 4 cooling schedules are shown in Tables 3 and 4 for examples 1 and 2 respectively. The initial acceptance ratio, χ_0, was set equal to 0.9 in all the implementations.

The highest value of t_0 was used by schedule # 2 which *was* decreased very rapidly as the algorithm progressed. The value of t_0 used in schedule # 8 is about twice the one used in schedule # 4, because of inclusion of an extended range in its computation.

Table 3. Results using example 1.

Schedule	Parameter Values	Initial Temperature	objective value	function Evaluations
1	λ=0.7	8217	484.0	30,808
	λ=0.25		438.75	30,337
2	δ=1.0	10,889	462. 89	9,665
	δ=5.0		466.59	12,401
3	*f*=0.9	290.7	449.5	38,753
	f=0.95		472.75	37,028
4	*a=2, f=.5*	643.7	425.25	18,992
	a=2, f=.3		469.0	20,691
	a=3, f=.25		431.812	20,956

Table 4. Results using Example 2.

Schedule	Parameter Values	Initial Temp.	Optimum Found	Function Evaluations
1	λ=0.7	305.97	491.46	2578
	λ=0.25		483.74	1640
2	δ=1.0	137.54	483.35	2873
	δ=5.0		483.35	3321
3	*f*=0.9	32.415	494.79	5900
	f=0.95		469.89	4672
	f=0.99		483.55	5869
4	*a=3, f=.5*	186.3	484.79	4680
	a=3, f=.25		485.24	3997
	a=2, f=.5		496.33	4434
	a=2, f=.4		483.55	4780

Table 5. Comparison of results among three methods of solving discrete variable truss optimization problem.

Member group no.	Starting design	Design by TS *	Design by GA[?]	Design by PS-SA**
1	1.0	0.1	0.1	0.1
2	3.4	0.6	1.8	0.4
3	3.4	3.4	2.3	3.4
4	1.0	0.1	0.2	0.1
5	3.4	1.8	0.1	2.0
6	2.0	0.9	0.8	1.0
7	3.4	0.4	1.8	0.4
8	3.4	3.4	3.0	3.4
Weight	969.01	481.523	546.01	481.33
Constr. Viol.	0.0	0.00083	0.00	0.00035

* TS – Tabu Search,[6],

[?] GA – Genetic Algorithm[7]

** PS-SA – Parametric Schedule-Simulated Annealing

Schedule # 1, executed with λ = 0.7 compared unfavorably with the one with a reduced value of λ = 0.25. A similar trend is proven, for both examples, when schedule # 3 was implemented. A strategy to dynamically determine δ based on problem behavior would be a significant improvement in the annealing technique. It is conjectured that the rapid decrement factor contributed towards locating better solutions.

Schedule # 4 is the one proposed here consistently found good results for both problems. Notably values of (*a, f*) = (2, 0.5) and (2, 0.4) generated better results for problems # 1 and # 2 respectively. Interestingly enough, when allowed to violate

constraints to the tune of < 0.001, each of the parameter set in schedule # 4 generated an optimum which compared favorably with the results reported by Dhingra and Bennage [8] and the those of Rajeev and Krishnamurthy [9] and listed in Table 5 for comparison. When allowed similar violation of constraints, nearly 2/3 of all other schedules found the same optimum (objective = 481.33), while the other 1/3 generated result very close to it. This is due to discrete nature of the variables.

The results of the present computational study show that most of the schedules are effective to the same order of accuracy and computational effort. Nevertheless, it is clearly observed that the schedules having control parameters can be tuned to their optimum levels by an appropriate manipulation of the parameters. Specially if the user has a good knowledge of the behavior of the problem.

5 Summary and Remarks

Cooling schedule is the control mechanism of annealing algorithms. The primary aspect in this paper is the introduction of a new adaptive cooling schedule called the 'Parametric Schedule.' According to this schedule, all parameters adapt themselves depending on the behavior of the problem at hand, except the temperature decrement rule that follows a Gaussian decrement function. Two example problems are used to compare the performance of parametric schedule with three other schedules from the literature. Complexity of cooling schedule did not correlate with quality of results. The proposed parametric schedule performed better than or competitively with most other schedules.

References

1. Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Vol 220, pp. 671–680,1983.
2. Aarts, E.H.L. and van Laarhoven, P.J.M., "Statistical Cooling: A General Approach to Combinatorial Optimization Problems," *Philips J. of Research*, Vol. 40, pp. 193–226, 1985.
3. Atiqullah, Mir, S.S. Rao, "Tuned Annealing for Optimization" Proceedings of the 2001 International Conference of Computational Science (ICCS2001), San Francisco, CA, May 28–30, 2001.
4. S.S. Rao, "Engineering Optimization: Theory and practice," Wiley 1995.
5. Huang, M.D., Romeo, F, and Sangiovanni-Vincentelli, A.L., "An Efficient General Cooling Schedule for Simulated Annealing," in Proceedings of IEEE International Conference on Computer-Aided Design, pp. 381–384, Santa Clara, November 1986.
6. Otten, Ralph H.J.M and van Ginneken, Lukas P.P.P., "Floorplan design using simulated annealing," Proceding of the IEEE International Conference on Computer Aided Design, Santa Clara, pp. 96–98., Nov. 1984.
7. Romeo, F. and Sangiovanni-Vincentelli, A.L., "Probabilistic Hill Climbing Algorithms: Properties and Applications," *Proceedings of the Chapel Hill Conference and VL SI*, pp.
8. W.A. Bennage, and A.K. Dhingra, "Optimization of Truss Topology Using Tabu Search," International Journal for Numerical Methods in Engineering, Vol. 38, pp. 4035–4052, 1995.
9. Rajeev, S., Krishnamurty, C.S., "Discrete Optimization of Structures using Genetic Algorithms," *Journal of Struct. Engineering*,Vol. 118, No.5, pp. 1233–1250, May 1992.

A Problem-Specific Convergence Bound for Simulated Annealing-Based Local Search

Andreas A. Albrecht

University of Hertfordshire
Dept. of Computer Science
Hatfield, Herts AL10 9AB, UK
A.Albrecht@herts.ac.uk

Abstract. We investigate the convergence of simulated annealing with emphasis on the probability $1-\delta$ to be in an optimum solution. The analysis is carried out for a logarithmic cooling schedule $c(k) = \Gamma/\ln(k+2)$, i.e., the temperature is lowered at any step k. We prove that after $k > (n/\delta)^{O(\Gamma)}$ steps the probability to be in an optimum solution is larger than $1-\delta$, where n is an upper bound for the size of local neighbourhoods. The parameter Γ is problem specific and depends on the underlying energy landscape. By counting the occurrences of configurations, we demonstrate for an application with known optimum solutions that the lower bound indeed ensures the stated probability for a relatively small constant in $O(\Gamma)$.

Keywords: Local Search, Markov Chains, Simulated Annealing, Cooling Schedules, Convergence.

1 Introduction

Simulated annealing was introduced independently by KIRKPATRICK et al. [8] and V. ČERNY [5] as a new class of algorithms computing approximate solutions of combinatorial optimisation problems. The general approach itself is derived from METROPOLIS' method [9] to calculate equilibrium states for substances consisting of interacting molecules.

Simulated annealing algorithms can be distinguished by the method that determines how the temperature is lowered at different times of the computation process (the so-called cooling schedule). If the temperature is kept constant for a (large) number of steps, one can associate a homogeneous Markov chain with this type of computation. Under some natural assumptions, the probabilities of configurations tend to the Boltzmann distribution for such homogeneous Markov chains (see [12,13,14]). This type of simulated annealing algorithms has been studied intensely and numerous heuristics have been devised for a wide variety of combinatorial optimisation problems (see [1,4,10,11,15]).

If the temperature is lowered at each step, the probabilities of configurations are computed by an inhomogeneous Markov chain. The special case of logarithmic cooling schedules has been investigated in [2,3,4,6]. B. HAJEK [6] proved

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 405–414, 2004.
© Springer-Verlag Berlin Heidelberg 2004

that logarithmic simulated annealing tends to an optimum solution if and only if the cooling schedule is lower bounded by $\Gamma/\ln(k+2)$, where Γ is the maximum value of the escape height from local minima of the underlying energy landscape.

Given the configuration space $\mathcal{F}$, let $\mathbf{a}_f(k)$ denote the probability to be in configuration f after k steps of an inhomogeneous Markov chain. The problem is to find a lower bound for k such that $\sum_{f\in\mathcal{F}_{\min}} \mathbf{a}_f(k) > 1-\delta$ for $f \in \mathcal{F}_{\min} \subseteq \mathcal{F}$ minimising the objective function. Let n denote a uniform upper bound for the number of neighbours of configurations $f \in \mathcal{F}$. We obtain a run-time of $k \geq (n/\delta)^{O(\Gamma)}$ to ensure that with probability $1-\delta$ the minimum value of the objective function has been approached.

The approach is illustrated by an example from machine learning: From a single positive example and a number of negative examples a conjunction of shortest length representing the examples has to be calculated. Counting of configurations demonstrates that the lower bound for k indeed ensures the stated probability for a relatively small constant in $O(\Gamma)$.

2 Preliminaries

The configuration space is *finite* and denoted by $\mathcal{F}$. We assume an objective function $\mathcal{Z} : \mathcal{F} \longrightarrow \mathbb{N}$ that for simplicity takes its values from the set of integers. By $\mathcal{N}_f$ we denote the set of neighbours of f, including f itself.

We assume that $\mathcal{F}$ is *reversible*: Any transition $f \to f'$, $f' \in \mathcal{N}_f$, can be performed in the reverse direction, i.e., $f \in \mathcal{N}_{f'}$, and we set

$$n := \max_{f\in\mathcal{F}} |\mathcal{N}_f| \,. \tag{1}$$

The set of minimal elements (optimum solutions) is defined by

$$\mathcal{F}_{\min} := \big\{\, f \,:\, \forall f'\big(f' \in \mathcal{F} \to\ \mathcal{Z}(f) \,\leq\, \mathcal{Z}(f')\big)\big\}.$$

Example. We consider Boolean conjunctive terms defined on n variables. The conjunctions are of length $\lceil \log n \rceil$ and have to be guessed from negative examples and one positive example $\tilde{\sigma}$. Hence, $\mathcal{F}$ consists of $f_{\tilde{\sigma}} = x^{\tilde{\sigma}}$ and all sub-terms f that can be obtained by deleting a literal from $x^{\tilde{\sigma}}$ in such a way that all negative examples are rejected. Each neighbourhood $\mathcal{N}_f$ contains at most $n' \leq n+1$ elements, since deleted literals can be included again. The configuration space is therefore reversible. The objective function is given by the length of terms f and in this case the optimum in known and equal to $\lceil \log n \rceil$.

In simulated annealing, the transitions between neighbouring elements are depending on the objective function $\mathcal{Z}$. Given a pair of configurations $[f, f']$, $f' \in \mathcal{N}_f$, we denote by $G[f,\, f']$ the probability of generating f' from f and by $A[f, f']$ the probability of accepting f' once it has been generated from f. Since we consider a single step of transitions, the value of $G[f,\, f']$ depends on the set $\mathcal{N}_f$. As in most applications of simulated annealing, we take a uniform probability which is given by

$$G[f,\, f'] := \frac{1}{|\mathcal{N}_f|}. \tag{2}$$

The acceptance probabilities $A[f,\, f']$, $f' \in \subseteq \mathcal{F}$ are derived from the underlying analogy to thermodynamic systems [1,9]:

$$A[f, f'] := \begin{cases} 1, & \text{if } \mathcal{Z}(f') - \mathcal{Z}(f) \leq 0, \\ e^{-\frac{\mathcal{Z}(f') - \mathcal{Z}(f)}{c}}, & \text{otherwise,} \end{cases} \tag{3}$$

where c is a control parameter having the interpretation of a *temperature* in annealing procedures.

Finally, the probability of performing the transition between f and f' is defined by

$$\mathbf{Pr}\{f \to f'\} = \begin{cases} G[f,\, f'] \cdot A[f,\, f'], & \text{if } f' \neq f, \\ 1 - \sum\limits_{f' \neq f} G[f,\, f'] \cdot A[f,\, f'], & \text{otherwise.} \end{cases} \tag{4}$$

By definition, the probability $\mathbf{Pr}\{f \to f'\}$ depends on the control parameter c.

Let $\mathbf{a}_f(k)$ denote the probability of being in the configuration f after k steps performed for the same value of c. The probability $\mathbf{a}_f(k)$ can be calculated in accordance with

$$\mathbf{a}_f(k) \;:=\; \sum_h \mathbf{a}_h(k-1) \cdot \mathbf{Pr}\{h \to f\}. \tag{5}$$

The recursive application of (5) defines a Markov chain of probabilities $\mathbf{a}_f(k)$, where $f \in \mathcal{F}$ and $k = 1, 2, \ldots$. If the parameter $c = c(k)$ is a constant c, the chain is said to be a *homogeneous* Markov chain; otherwise, if $c(k)$ is lowered at any step, the sequence of probability vectors $\mathbf{a}(k)$ is an *inhomogeneous* Markov chain.

In the present paper we are focusing on a special type of inhomogeneous Markov chains where the value $c(k)$ changes in accordance with

$$c(k) \;=\; \frac{\Gamma}{\ln(k \,+\, 2)},\; k \,=\, 0,\, 1,\, \ldots. \tag{6}$$

The choice of $c(k)$ is motivated by HAJEK's Theorem [6] on logarithmic cooling schedules for inhomogeneous Markov chains:

Theorem 1 [6] *Under some natural assumptions, the asymptotic convergence* $\sum_{f \in \mathcal{F}_{\min}} \mathbf{a}_f(k) \underset{k \to \infty}{\longrightarrow} 1$ *of the stochastic algorithm defined by (3), (4), and (6) is guaranteed if and only if* $\Gamma \geq \max_{g_{\min}} \text{height}(g_{\min})$, *where* $\text{height}(g_{\min})$ *is the minimum escape height of* $g_{\min} \in \mathcal{F}_{\min}$.

Let K_0 denote the maximum of the minimum number of transitions to reach an optimum solution starting from an arbitrary $f \in \mathcal{F}$. In Section 3, we will prove the following convergence result:

Theorem 2 *Given the configuration space* $\mathcal{F}$ *and the parameter* Γ *as defined in Theorem 1, then* $k \geq \max\{K_0, (n/\delta)^{O(\Gamma)}\}$ *implies for arbitrary initial probability distributions* $\mathbf{a}(0)$ *the relation*

$$\sum_{\tilde{f} \notin \mathcal{F}_{\min}} \mathbf{a}_{\tilde{f}}(k) < \delta, \quad \textit{and therefore,} \quad \sum_{f' \in \mathcal{F}_{\min}} \mathbf{a}_{f'}(k) \;\geq\; 1 \,-\, \delta.$$

3 Convergence Analysis

We introduce the following partition of the set of configurations with respect to the value of the objective function:
$L_0 \quad := \mathcal{F}_{\min}$ and
$L_{h+1} := \{f : f \in \mathcal{F} \wedge \forall f'\big(f' \in \mathcal{F} \setminus \bigcup_{i=0}^{h} L_i \rightarrow \mathcal{Z}(f') \geq \mathcal{Z}(f)\}$.
For any particular element $f \in \mathcal{F}$, we introduce notations for the number of neighbours with a certain length. We recall that the definition of the neighbourhood relation implies that $\mathcal{N}_f$ contains only one element of length $l(f)$ - the element f itself. We denote

$$s(f) := |\{\, f' \,:\, f' \in \mathcal{N}_f \wedge \mathcal{Z}(f') \,>\, \mathcal{Z}(f)\}|, \tag{7}$$
$$r(f) := |\{\, f' \,:\, f' \in \mathcal{N}_f \wedge \mathcal{Z}(f') \,<\, \mathcal{Z}(f)\}|\,. \tag{8}$$

Thus, from the definition of $\mathcal{N}_f$ we have

$$s(f) \,+\, r(f) = |\mathcal{N}_f| \;-1. \tag{9}$$

We consider the probability $\mathbf{a}_f(k)$ to be in the configuration $f \in \mathcal{F}$ after k transitions of an *inhomogeneous* Markov chain that is defined in accordance with (6). We observe that for $\mathcal{Z}(f') > \mathcal{Z}(f)$ the acceptance probability (3) can be rewritten as

$$e^{-(\mathcal{Z}(f')-\mathcal{Z}(f))/c(k)} = \frac{1}{\big(k+2\big)^{(\mathcal{Z}(f')-\mathcal{Z}(f))/\Gamma}},\; k \geq 0. \tag{10}$$

To simplify notations, we define a new objective function where we maintain the same notation: $\mathcal{Z}(f) := \mathcal{Z}(f)/\Gamma$. This is possible because Γ is a constant value.

3.1 A Parameterized Expansion of Probabilities

In (5), we separate the probabilities according to whether or not f' equals f, and we obtain:

Lemma 1 *The value of* $\mathbf{a}_f(k)$ *can be calculated from probabilities of the previous step by*

$$\mathbf{a}_f(k) = \left(\frac{s(f)+1}{|\mathcal{N}_f|} - \sum_{i=1}^{s(f)} \frac{\big(k+1\big)^{-(\mathcal{Z}(f_i)-\mathcal{Z}(f))}}{|\mathcal{N}_f|}\right) \cdot \mathbf{a}_f(k-1) + \sum_{i=1}^{s(f)} \frac{\mathbf{a}_{f_i}(k-1)}{|\mathcal{N}_{f_i}|} +$$
$$+ \sum_{j=1}^{r(f)} \frac{\mathbf{a}_{f_j}(k-1)}{|\mathcal{N}_{f_j}|} \cdot \frac{1}{\big(k+1\big)^{\mathcal{Z}(f)-\mathcal{Z}(f_j)}}.$$

The representation (expansion) from Lemma 1 will be used in the following as the main relation reducing $\mathbf{a}_f(k)$ to probabilities from previous steps. Besides taking into account the value of the objective function in classes L_h, the elements of the

configuration space are distinguished additionally by their minimum distance to $\mathcal{F}_{\min}$: Given $f \in \mathcal{F}$, we consider a shortest path of length $\mathrm{dist}(f)$ with respect to neighbourhood transitions from f to $\mathcal{F}_{\min}$. We introduce a partition of $\mathcal{F}$ in accordance with $\mathrm{dist}(f)$:

$$f \in M_i \Longleftrightarrow \mathrm{dist}(f) = i \geq 0, \quad \text{and} \quad \mathcal{M}_{\mathrm{d_m}} = \bigcup_{i=0}^{\mathrm{d_m}} M_i, \tag{11}$$

where $M_0 := L_0 = \mathcal{F}_{\min}$ and d_m is the maximum distance. Thus, we distinguish between distance levels M_i related to the minimum number of transitions required to reach an element of $\mathcal{F}_{\min}$ and the levels L_h that are defined by the objective function.

Since we want to analyze the convergence to elements from $M_0 = L_0 = \mathcal{F}_{\min}$ we have to show that the value

$$\sum_{f \notin M_0} \mathbf{a}_f(k) \tag{12}$$

becomes small for large k. We suppose that $k \geq d_m$, and we are going backwards from the k^{th} step. We consider the expansion of a particular probability $\mathbf{a}_f(k)$ as shown in Lemma 1. At the same step k, the neighbours of f are generating terms containing $\mathbf{a}_f(k-1)$ as a factor, in the same way, as $\mathbf{a}_f(k)$ generates terms with factors $\mathbf{a}_{f_i}(k-1)$ and $\mathbf{a}_{f_j}(k-1)$ in Lemma 1. If we consider the entire sum $\sum_{f \notin M_0} \mathbf{a}_f(k)$, the terms corresponding to a particular $\mathbf{a}_f(k-1)$ can be collected together to form a single term.

Firstly, we consider $f \in M_i$, $i \geq 2$. In this case, f does not have neighbours from M_0, i.e., the expansion from Lemma 1 appears for all neighbours of f in the reduction of $\sum_{f \notin M_0} \mathbf{a}_f(k)$ to step $(k-1)$. Therefore, taking all terms together that contain $\mathbf{a}_f(k-1)$, we obtain

$$\begin{aligned} &\mathbf{a}_f(k-1) \cdot \left\{ \left(\frac{|\mathcal{N}_f| - r(f)}{|\mathcal{N}_f|} - \sum_{i=1}^{s(f)} \frac{1}{|\mathcal{N}_f|} \cdot \frac{1}{\big(k+1\big)^{\mathcal{Z}(f_i)-\mathcal{Z}(f)}} \right) + \right. \\ &\left. + \sum_{i=1}^{s(f)} \frac{1}{|\mathcal{N}_f|} \cdot \frac{1}{\big(k+1\big)^{\mathcal{Z}(f_i)-\mathcal{Z}(f)}} + \sum_{j=1}^{r(f)} \frac{1}{|\mathcal{N}_f|} \right\} \\ &= \mathbf{a}_f(k-1). \end{aligned} \tag{13}$$

Secondly, if $f \in M_1$, the neighbours from M_0 are missing in $\sum_{f \notin M_0} \mathbf{a}_f(k)$ at the step to $(k-1)$, i.e., they do not generate terms containing probabilities from higher levels. For $f' \in M_0$, the expansion from Lemma 1 contains the terms $\mathbf{a}_{f_i}(k-1)/\,|\,\mathcal{N}_{f_i}\,|$ for $f_i \in M_1$ (and there are no terms for f_j with a smaller value of the objective function since $f' \in M_0$). Thus, the terms $\mathbf{a}_{f_i}(k-1)/\,|\,\mathcal{N}_{f_i}\,|$ are not available for $f = f_i \in M_1$ in the reduction of $\sum_{f \notin M_0} \mathbf{a}_f(k)$ to step $(k-1)$, when one tries to establish a relation like (13) for elements of M_1. For each $f \in M_1$, there are $r(f)$ such terms related to neighbours from M_0, see (8).

Therefore, in the expansion of $\sum_{f \notin M_0} \mathbf{a}_f(k)$, the following arithmetic term is generated when the particular f is from M_1:

$$\Big(1 - \frac{r(f)}{|\mathcal{N}_f|}\Big) \cdot \mathbf{a}_f(k-1). \tag{14}$$

We introduce the following abbreviations:

$$\varphi(f', f, v) := \frac{\big(k+2-v\big)^{-(\mathcal{Z}(f')-\mathcal{Z}(f))}}{|\mathcal{N}_f|};\; D_f(k-v) := \frac{s(f)+1}{|\mathcal{N}_f|} - \sum_{i=1}^{s(f)} \varphi(f', f, v). \tag{15}$$

The diminishing factor $\big(1 - r(f)/|\mathcal{N}_f|\big)$ appears by definition for all elements of M_1. At subsequent reduction steps, the factor is "transmitted" successively to all probabilities from higher distance levels M_i because any element of M_i has at least one neighbour from M_{i-1}. The main task is now to analyze how this diminishing factor changes when it is transmitted to higher distance levels. We denote

$$\sum_{f \notin M_0} \mathbf{a}_f(k) = \sum_{f \notin M_0} \mu(f, v) \cdot \mathbf{a}_f(k-v) + \sum_{f' \in M_0} \mu(f', v) \cdot \mathbf{a}_{f'}(k-v), \tag{16}$$

i.e., the coefficients $\mu(\tilde{f}, v)$ are the factors at probabilities after v steps of an expansion of $\sum_{f \notin M_0} \mathbf{a}_f(k)$. We establish a recursive relation for the coefficients $\mu(\tilde{f}, v)$ from (16), where we use the same method as in (13). We denote for neighbouring elements $f' < \tilde{f}$, if $\mathcal{Z}(f') < \mathcal{Z}(\tilde{f})$, and $f' > \tilde{f}$ for the reverse relation of the objective function. Taking into account (15), we obtain

$$\begin{aligned} \mu(\tilde{f}, v) = \mu(\tilde{f}, v-1) \cdot D_{\tilde{f}}(k-v) + \sum_{f' < \tilde{f}} \frac{\mu(f', v-1)}{|\mathcal{N}_{\tilde{f}}|} + \\ + \sum_{f'' > \tilde{f}} \mu(f'', v-1) \cdot \varphi(f'', \tilde{f}, v). \end{aligned} \tag{17}$$

The following simple transformation allows us to consider probabilities for $k \geq \widehat{k}$ only: If $f \notin M_0$, we take $\nu(f, v) = 1 - \mu(f, v)$ instead of $\mu(f, v)$; for $f' \in M_0$ we continue to use $\mu(f', v)$. When $\mu(f, v)$ is substituted in (17) by $1 - \nu(f, v)$, we obtain the same relation for $\nu(f, v)$ because the sum of transition probabilities equals 1 within the neighbourhood $\mathcal{N}_f$.

The term $r(f)/|\mathcal{N}_f|$ appears in all recursive equations of $\nu(f, v)$, $f \in M_1$ and $v \geq 1$, and the same is valid for the value $\sum_{f > f'} \varphi(f, f', v)$ in all $\mu(f', v)$, $f' \in M_0$. Therefore, all arithmetic terms T defining $\nu(f, v)$ and $\mu(f', v)$ are derived from terms of the type $r(f)/|\mathcal{N}_f|$ and $\sum_{f > f'} \varphi(f, f', v)$. We keep track of each individual term that is generated by a recursive step as given in (17). For this purpose, the coefficients $\nu(f, v)$ are represented by a sum $\sum_i T_i$ of arithmetic terms.

Definition 1 *The terms* $r(f)/|\mathcal{N}_f|$, $f \in M_1$, *and* $\sum_{f > f'} \varphi(f, f', v)$, $f' \in M_0$, *are called* source terms *of* $\nu(f, v)$ *and* $\mu(f', v)$, *respectively, where* $v \geq 1$.

During an expansion of $\sum_{f\notin M_0}\mathbf{a}_f(k)$ backwards according to (16), the source terms are distributed permanently to higher distance levels M_j as well as to elements from M_0. That means, in the same way as for M_1, the calculation of $\nu(f,v)$ ($\mu(f',v)$ for M_0) is repeated almost identically at any step, only the "history" of generations becomes longer. We introduce a counter $\mathbf{r}(f)$ to terms T that indicates the step at which the term has been generated from source terms. The value $\mathbf{r}(f)$ is called the rank of a term and we set $\mathbf{r}(f)=1$ for source terms T from Definition 1. Basically, the rank $\mathbf{r}(f)\geq 1$ indicates the number of factors when T is represented by the subsequent multiplications according to the recurrent generation rule (17). Let $\mathcal{T}_j(\tilde{f},v)$ be the set of j^{th} rate arithmetic terms from $\nu(\tilde{f},v)$ with the same rank $\mathbf{r}(f)$, where $\tilde{f}\in\mathcal{M}_{d_m}\backslash M_0$. We set

$$\mathbf{S}_j(\tilde{f},v) := \sum_{T\in\mathcal{T}_j(\tilde{f},v)} T. \tag{18}$$

The same notation is used in case of $f'=\tilde{f}\in M_0$ with respect to $\mu(f',v)$. Now, the coefficients $\nu(\tilde{f},v)$, $\mu(f',v)$, can be represented by

$$\nu(\tilde{f},v)=\sum_{j=1}^{v}\mathbf{S}_j(\tilde{f},v) \quad \text{and} \quad \mu(f',v) \;=\; \sum_{j=1}^{v}\mathbf{S}_j(f',v). \tag{19}$$

We compare the computation of $\nu(f,v)$ and $\mu(f',v)$ for two different values $v=k_1$ and $v=k_2$, i.e., $\nu(f,v)$ is calculated backwards from k_1 and k_2, respectively. Let $\mathbf{S}_j^1$ and $\mathbf{S}_j^2$ denote the corresponding sums of terms related to two different starting steps k_1 and k_2. From Definition 1 we see that the source term $r(f)/|\mathcal{N}_f|$ does not depend on k. For the second type of source terms, we employ the simple equation $k_2-(k_2-k_1+v)=k_1-v$ and obtain

Lemma 2 *Given $k_2\geq k_1\geq K_0$ and $1\leq j\leq k_1$, then for each $f\in\mathcal{M}$:*

$$\mathbf{S}_j^1(f,v)=\mathbf{S}_j^2(f,k_2-k_1+v).$$

Now, we have from (16):

$$\begin{aligned}\sum_{f\notin M_0}\mathbf{a}_f(k_1) &= \sum_{f\notin M_0}\big(\mathbf{a}_f(k_1)-\mathbf{a}_f(k_2)\big)+\sum_{f\notin M_0}\mathbf{a}_f(k_2) \\ &= \sum_{f\notin M_0}\big(\nu(f,k_2-k_1)-\nu(f,0)\big)\cdot\mathbf{a}_f(k_1)+ \\ &\quad+\sum_{f'\in M_0}\big(\mu(f',0)-\mu(f',k_2-k_1)\big)\cdot\mathbf{a}_{f'}(k_1)+\sum_{f\notin M_0}\mathbf{a}_f(k_2).\end{aligned} \tag{20}$$

Lemma 2 and (19) imply

$$\sum_{f\notin M_0}\big(\nu(f,k_2-k_1)-\nu(f,0)\big)\cdot\mathbf{a}_f(k_1)=\sum_{f\notin M_0}\sum_{j=1}^{k_2-k_1}\mathbf{S}_j^2(f,k_2-k_1)\cdot\mathbf{a}_f(k_1). \tag{21}$$

The same applies to $f'\in M_0$.

3.2 Upper Bounds for Elementary Expressions

To find upper bounds for (21), we estimate $\mathbf{a}_f(k_1)$ for configurations different from global and local minima, and the $\mathbf{S}_j^2(f, k_2 - k_1)$ are then estimated for global and local minima separately. To distinguish between the two cases is necessary since for small j and f different from global and local minima, the values $\mathbf{S}_j^2(f, k_2 - k_1)$ are relatively large (cf. Definition 1). We note that the recursive application of (17) generates negative summands in the representation of values $\mathbf{S}_j(f, v)$, and we note that negative and positive products can be considered separately at all distance levels. We set $\mathbf{S}_j(f, v) = \mathbf{S}_j^+(f, v) - \mathbf{S}_j^-(f, v)$ and we concentrate on upper bounds of $\mathbf{S}_j^+(f, v)$ only. To simplify notations, we use $\mathbf{S}_j(f, v)$ instead of $\mathbf{S}_j^+(f, v)$. Furthermore, we use instead of $n + 1$ from $N_{\tilde{f}} \leq n + 1$ (see (1)) the value $n' = n + 1$, and for convenience n again for n'. We set $\widehat{\mathcal{M}} := \{f : r(f) \geq 1\}$, and we can prove

$$\sum_{f \in L_h \cap \widehat{\mathcal{M}}} \mathbf{a}_f(k) < \frac{2 \cdot (h_{\max} - h) \cdot n^3}{(k + 2 - n^3)^\gamma}. \tag{22}$$

Now, we estimate $\mathbf{S}_j(t, v)$ specifically for local and global minima. Here, we use the property that backwards expansions "entering" a local or global minimum are multiplied by $1/(k + 2 - v)^\gamma$, i.e., the upper bound is of the the type $\Pi/(k + 2 - v)^\gamma$, where Π represents the sum of products leading from M_1 (or M_0) to the local or global minimum. From (15) and (17) we conclude

$$\mathbf{S}_j(f, v) = \sum \Phi_1 \cdots \Phi_j \ \leq \sum_{[d,g,h_1,h_2]} \ \sum_{\text{Positions of } D,\ldots,H_2} D^d \cdot G^g \cdot H_1^{h_1} \cdot H_2^{h_2}, \tag{23}$$

$d + g + h_1 + h_2 = j$, where D is the probability to stay in a local minimum, G corresponds to steps decreasing the objective function (we recall, that we are going backwards in the expansion of $\mathbf{a}_f(k - v)$), H_1 is associated with steps increasing the objective function, and H_2 is from the probability to stay in the same configuration which is not a local minimum.

In the following, we utilise the fact that the objective function changes only by ± 1 during one transition. However, the proposition can be formulated in a similar way for general types of objective functions.

Lemma 3 *Let $S := \max_f \mathcal{Z}(f)$ denote the maximum value of the objective function. Then the numbers of steps g and h_1, where h_1 relates to factors H_1, must satisfy $\mid g - h_1 \mid \leq S$.*

For an upper bound of $\sum_{[d,g,h_1,h_2]} D^d \cdot G^g \cdot H_1^{h_1} \cdot H_2^{h_2}$, we need the following, relatively tight bound:

Lemma 4 *For $\Gamma \geq 2$, $k \geq 1$, $0 \leq a < k$, and $c = 1 + (k - a)/((\Gamma - 1) \cdot k)$, the following inequality holds:*

$$\sum_{v_1 \cdots v_a} \frac{1}{(k + 2 - v_1)^\gamma \cdots (k + 2 - v_a)^\gamma} < \binom{k}{a} \cdot \frac{c^a}{\big(k + 2\big)^{a \cdot \gamma}}. \tag{24}$$

Based on (21), Lemma 3, and Lemma 4 we derive an upper bound for $\sum \Phi_1 \cdot \Phi_2 \cdots \Phi_j$ from (23), which leads to

$$\sum_{f \in \mathcal{M} \setminus \widehat{\mathcal{M}}} \sum_{j=1}^{k_2-k_1} \mathbf{S}_j^2(f, k_2 - k_1) \cdot \mathbf{a}_f(k_1) < \frac{n^2}{(k+2-n)^\gamma} \cdot \sum_{f \in \mathcal{M} \setminus \widehat{\mathcal{M}}} \mathbf{a}_f(k_1). \quad (25)$$

Now, (22) and (25) are used to prove (we note that $\sum_{f \in \mathcal{F}} \mathbf{a}_f(k_1) = 1$)

$$|\sum_{f \notin M_0} \big(\nu(f, k_2) - \nu(f, k_1)\big) \cdot \mathbf{a}_f(k_1)| < \frac{O(n^4)}{(k+2-n)^\gamma}, \quad (26)$$

where we assume $h_{\max} =)(n)$. Here, we considered $\mathbf{S}_j^+$, and (21) has been applied to these values only, but the same holds for $\mathbf{S}_j^-$. Thus, we can derive in the same way the corresponding upper bound for $\big(\mu(f', k_1) - \mu(f', k_2)\big)$.

Proof of Theorem 2: We employ the representation from (20) for $k = k_1$. We utilize Theorem 1, i.e., if the constant Γ from (6) is sufficiently large, the inhomogeneous simulated annealing procedure defined by (2), (3), and (4) tends to the global minimum of $\mathcal{Z}$ on $\mathcal{F}$. The value k_2 from (26) is larger but independent of $k_1 = k$, i.e., we can take a $k_2 > k$ such that

$$\sum_{\tilde{f} \notin M_0} \mathbf{a}_{\tilde{f}}(k_2) < \frac{\delta}{3}.$$

We obtain the stated inequality, if additionally both differences $\sum_{\tilde{f} \notin M_0} \big(\nu(\tilde{f}, k_2) - \nu(\tilde{f}, k)\big)$ and $\sum_{f' \in M_0} \big(\mu(f', k) - \mu(f', k_2)\big)$ are smaller than $\delta/3$, and taking into account (26), we finally arrive at

$$k > \big(\frac{n}{\delta}\big)^{O(\Gamma)} \geq n - 2 + \big(\frac{3 \cdot O(n^4)}{\delta}\big)^\Gamma.$$

q.e.d.

4 Computational Experiments

We are given a set $\mathcal{S} \subseteq \{0,1\}^n$ of uniformly distributed binary n-tuples $\tilde{\eta} = \eta_1 \cdots \eta_n$ that represent negative examples for an unknown target conjunction $C_\ell = x_{i_1}^{\sigma_{i_1}} \& x_{i_2}^{\sigma_{i_2}} \& \cdots \& x_{i_\ell}^{\sigma_{i_\ell}}$ (here, we use $x^1 \equiv x$ and $x^0 \equiv \overline{x}$, i.e., $x^0 = 1$ for $x = 0$, and $x^0 = 0$ for $x = 1$), and a single positive example $\tilde{\sigma} = \sigma_1 \cdots \sigma_n$: $C_\ell(\tilde{\sigma}) = 1$ and $\forall \tilde{\eta} (\tilde{\eta} \in \mathcal{S} \to C_\ell(\tilde{\eta}) = 0)$. The task is to find a conjunction C_l of length $l \leq \ell$ that matches all of the samples, i.e., from C_ℓ generating the samples we do know only the length ℓ; cf. [7] and the Example in Section 2.

As explained in Section 2, we have $\Gamma \leq \lceil \log n \rceil$ for the problem to find a conjunction of length $\ell = \lceil \log n \rceil$. We implemented the search procedure for $m = 32$ negative examples, and for each element of $\mathcal{F}$ we counted the number of occurrences during the search procedure, in particular, for $\mathcal{F}_{\min}$. The calculations were repeated three times, and we present the average values (we obsereved only small deviations). The constant c in $O(\Gamma) = c \cdot \Gamma$ was set to $c = 1$.

		$n = 8$ and $\Gamma = 3$		$n = 16$ and $\Gamma = 4$	
δ	$1 - \delta$	k according to Theorem 2 ($c = 1$)	Frequency of $f \in \mathcal{F}_{\min}$	k according to Theorem 2 ($c = 1$)	Frequency of $f \in \mathcal{F}_{\min}$
0.50	0.50	4096	0.739	1048576	0.786
0.25	0.75	32768	0.812	16777216	0.895
0.10	0.90	512000	0.945	655360000	0.953
0.01	0.99	512000000	0.996	———	——-

Frequencies of $f \in \mathcal{F}_{\min}$.

As we can see, the experimental results are in compliance with Theorem 2 for the small constant $c = 1$.

References

1. E.H.L. Aarts and J.H.M. Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach*, Wiley & Sons, New York, 1989.
2. S. Azencott (editor). *Simulated Annealing: Parallelization Techniques.* Wiley & Sons, New York, 1992.
3. O. Catoni. Rough Large Deviation Estimates for Simulated Annealing: Applications to Exponential Schedules. *Annals of Probability*, 20(3):1109–1146, 1992.
4. O. Catoni. Metropolis, Simulated Annealing, and Iterated Energy Transformation Algorithms: Theory and Experiments. *J. of Complexity*, 12(4):595–623, 1996.
5. V. Černy. A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm. Preprint, Inst. of Physics and Biophysics, Comenius Univ., Bratislava, 1982 (see also: *J. Optim. Theory Appl.*, 45:41–51, 1985).
6. B. Hajek. Cooling Schedules for Optimal Annealing. *Mathem. Oper. Res.*, 13:311–329, 1988.
7. M. Kearns, M. Li, L. Pitt, and L.G. Valiant. Recent Results on Boolean Concept Learning. In *Proc. 4^{th} Int. Workshop on Machine Learning*, pp. 337–352, 1987.
8. S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.
9. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *J. of Chemical Physics*, 21(6):1087–1092, 1953.
10. S. Rajasekaran and J.H. Reif. Nested Annealing: A Provable Improvement to Simulated Annealing. *J. of Theoretical Computer Science*, 99(1):157–176, 1992.
11. F. Romeo and A. Sangiovanni-Vincentelli. A Theoretical Framework for Simulated Annealing. *Algorithmica*, vol. 6, no. 3, pp. 302–345, 1991.
12. E. Seneta. *Non-negative Matrices and Markov Chains.* Springer-Verlag, New York, 1981.
13. A. Sinclair and M. Jerrum. Approximate Counting, Uniform Generation, and Rapidly Mixing Markov Chains. *Information and Computation*, 82:93–133, 1989.
14. A. Sinclair and M. Jerrum. Polynomial-Time Approximation Algorithms for the Ising Model. *SIAM J. Comput.*, 22(5):1087–1116, 1993.
15. G. Sorkin. Efficient Simulated Annealing on Fractal Energy Landscapes. *Algorithmica*, 6:367–418, 1991.

Comparison and Selection of Exact and Heuristic Algorithms*

Joaquín Pérez O.[1], Rodolfo A. Pazos R.[1], Juan Frausto S.[2], Guillermo Rodríguez O.[3], Laura Cruz R.[4], and Héctor Fraire H.[4]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)
AP 5-164, Cuernavaca, Mor. 62490, México
{jperez,pazos}@sd-cenidet.com.mx
[2] ITESM, Campus Cuernavaca, México
AP C-99 Cuernavaca, Mor. 62589, México
juan.frausto@itesm.mx
[3] Instituto de Investigaciones Eléctricas, IIE
gro@iie.org.mx
[4] Instituto Tecnológico de Ciudad Madero, México
{lcruzreyes,hfraire}@prodigy.net.mx

Abstract. The traditional approach for comparing heuristic algorithms uses well-known statistical tests for meaningfully relating the empirical performance of the algorithms and concludes that one outperforms the other. In contrast, the method presented in this paper, builds a predictive model of the algorithms behavior using functions that relate performance to problem size, in order to define dominance regions. This method generates first a representative sample of the algorithms performance, then using a common and simplified regression analysis determines performance functions, which are finally incorporated into an algorithm selection mechanism. For testing purposes, a set of same-class instances of the database distribution problem was solved using an exact algorithm (Branch&Bound) and a heuristic algorithm (Simulated Annealing). Experimental results show that problem size affects differently both algorithms, in such a way that there exist regions where one algorithm is more efficient than the other.

1 Introduction

In the solution of many difficult combinatorial problems (such as the data distribution problem), exact and heuristic algorithms have been used. Exact algorithms have been extensively studied and are considered adequate for moderately size instances, whereas heuristic algorithms are considered promising for very large instances [1]. To get the best of both, it is necessary to analytically determine for what problem size it is convenient to use an exact algorithm and when it is better to use a heuristic algorithm. However the lack of mathematical

* This research was supported in part by CONACYT and COSNET.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 415–424, 2004.
© Springer-Verlag Berlin Heidelberg 2004

methods to predict the performance of these algorithms, especially the heuristic ones, hinders their evaluation and the selection of the best for a given instance [2].

The theoretical study of heuristic algorithm performance based on the average and worst cases, is generally difficult to carry out. Furthermore, since it describes algorithm performance on the limit, it does not help for determining how well the algorithms perform with specific instances [3]. On the other hand, experimental analysis is certainly adequate for specific instances and is widely reported in scientific papers. However, it is very informally used and does not satisfy minimal reproduction standards; furthermore, results obtained through rigorous statistical methods are seldom reported [4].

Experimental studies that compare exact and heuristic algorithms are scarce and of limited practical utility. The most complete and up-to-date work is the one presented by Hoos and Stutzle in [5], which focuses on the study of some of the best well-known algorithms for the satisfability problem, contrasting the performance of the exact and heuristic algorithms on a wide range of problem instances. A similar study presented in [6] proposes a very general technique to model the growth of the search cost with respect to the problem size. These works have contributed to improve the knowledge of the performance of specific algorithms on some classes of instances. However, for practical effects it becomes still necessary to have a mechanism that allows to select from several algorithms the one that solves the problem in the smallest available time.

In this paper a new method is proposed for (1) statistically evaluating and characterizing exact and heuristic algorithms using performance functions, which relate the solution quality and processing time with the problem size; and (2) choosing the most adequate algorithm for a specific instance. Experimental results corroborate that problem size is an important factor that affects algorithm performance, since there exist regions where each algorithm is more efficient than the other.

This paper is organized as follows. The problem application, used to prove our hypothesis is presented in Section 2, solution algorithms are described too. Section 3 describes a general method for comparison and selection of exact and heuristic algorithms, used as solution algorithms. The experimental results of applying, the proposed selection method, to the solution of the distribution design problem, are described in section 4.

2 The Distribution Design Problem

In this section the mathematical formulation of the data distribution problem modeled by DFAR (distribution, fragmentation, allocation and reallocation) and the solution algorithms are described.

2.1 DFAR Mathematical Model

DFAR is an integer (binary) programming model. In this model, the decision about storing an attribute m in site j is represented by a binary variable x_{mj}. Thus $x_{mj} = 1$ if m is stored in j, and $x_{mj} = 0$ otherwise.

The objective function below models costs using four terms: transmission, access to several fragments, fragment storage, and fragment migration.

$$\min z = \sum_k \sum_i f_{ki} \sum_m \sum_j q_{km} l_{km} c_{ij} x_{mj} + \sum_i \sum_k \sum_j c_1 f_{ki} y_{kj} + \sum_j c_2 w_j$$

$$+ \sum_m \sum_i \sum_j a_{mi} c_{ij} d_m x_{mj}. \quad (1)$$

where

f_{ki} = emission frequency of query k from site i;
q_{km} = usage parameter, $q_{km} = 1$ if query k uses attribute m, otherwise $q_{km} = 0$;
l_{km} = number of packets for transporting attribute m for query k;
c_{ij} = communication cost between sites i and j;
c_1 = cost for accessing several fragments to satisfy a query;
y_{kj} = indicates if query k accesses one or more attributes located at site j;
c_2 = cost for allocating a fragment to a site;
w_j = indicates if there exist attributes at site j;
a_{mi} = indicates if attribute m was previously located at site i;
d_m = number of packets for moving attribute m to another site if necessary;

The model solutions are subject to five constraints: each attribute must be stored in one site only, each attribute must be stored in a site which executes at least one query that uses it, variables w_j and y_{kj} are forced to adopt values compatible with those of x_{mj}, and site storage capacity must not be exceeded by the attributes stored at each site. The detailed description of this model can be found at [7].

2.2 Solution Algorithms

Since the distribution problem modeled by DFAR is NP-complete [8], a heuristic method is needed. As an exact solution method, the Branch&Bound algorithm implemented in the Lindo 6.01 commercial software was used. As a heuristics method, a variation of the Simulated Annealing algorithm, known as Threshold Accepting, was implemented. In the cases reported in the specialized literature, this version consumes less computing time and generates better quality solutions [9]. More details of the implementations are reported in [10].

3 Evaluation of Algorithms

In this section a statistical method is presented for comparing exact and heuristic algorithms. Additionally, the steps for estimating algorithm performance and selecting the best are detailed.

3.1 Method for Comparison of Exact and Heuristic Algorithms

The following method was devised for comparing exact and heuristic algorithms:

Step 1. Sampling. Obtain through experimentation a tabular description of each algorithm behavior for instances of different size. Behavior examples are the deviation from the optimum and processing time.

Step 2. Estimation. Find the estimation functions for the algorithm performance, by applying to the tabular results a statistical treatment based on approximation techniques (for example regression analysis).

Step 3. Algorithm Selection. Choose the best algorithm according to the problem size using a selection algorithm based on the performance functions and the user requirements.

3.2 Measuring Performance

For the different solution methods we use two comparison aspects: CPU time and error percentage. We use the CPU time to measure the processing cost, while for the quality aspect we use the percentage of deviation from the optimum. Both quantities were determined for instances of the same class, each one of different size. Unlike many works, we measure the size of the problem counting the bytes occupied by all the problem parameters. We consider that this measure is more exact because it has a total representativeness of the data structure.

In other researches, oriented to improve algorithms, used as comparison strategy: a) run the algorithm until obtaining a very near solution to the optimal, b) run the algorithm until a predetermined time [5]. However, our work is focused to the characterization of algorithms, for selection purpose, without seeking to modify them (they are black boxes). In these circumstances, we let operate the algorithms until their own termination criteria are satisfied.

3.3 Finding Performance Estimation Functions

As mentioned before, for the performance characterization only two aspects will be considered: quality of results and computational effort required. In the sequel, the performance functions will be denoted by $T(n)$ and $E(n)$. The first one is the efficiency function which represents the relationship between problem size and the processing time of a specific algorithm, whereas the second one is the efficacy function which represents the relationship between problem size and the quality of the solution found by the algorithm. The quality is determined by the deviation from the optimum.

For test purposes a common and simplified method based on regression analysis was used. In this method n is the independent variable that represents the problem size, t is the random variable that represents the algorithm processing time, $\bar{t}$ is the average time, and r is the sample size.

Step 1. Generate a set of feasible polynomials, i.e. those whose degree g is small and smaller than the sample size.

$$E(n) = a_0 + a_1 n + a_2 n^2 + \ldots + a_g n^g \qquad 1 \leq g \leq (r-1)$$
$$T(n) = b_0 + b_1 n + b_2 n^2 + \ldots + b_g n^g \qquad 1 \leq g \leq (r-1)$$

Step 2. Calculate the coefficients a and b of the set of feasible polynomials using a fitness method such as least squares.

Step 3. Select the most adequate polynomial using statistical tests, which quantify their goodness to represent the relationship between performance and problem size. In order to increase the confidence level of the chosen function, three fit tests are recommended: estimation of the error variance, the global F test, and the Student $\mathbf{t}$ test. The first provides a preliminary assessment of the function confidence, with the second a subset of useful functions is obtained, and with the third the usefulness of the candidate function coefficients is determined. Table 1 presents the equations and conditions that are used to determine the goodness of fit of the efficiency polynomials, which are similar for the efficacy polynomials.

Table 1. Goodness tests

Error Variance	Global F Test	$\mathbf{t}$ Student Test
$se^2 = \frac{\sum_i^r (t_i - T(n_i))^2}{r-(g+1)}$	$R^2 = 1 - \frac{\sum_i^r (t_i - T(n_i))^2}{\sum_i^r (t_i - t)^2}$	se_{b_i} = standard error of b_i calculated by least squares
The polynomial is adequate if it has the smallest se value.	$F = \frac{R^2/g}{(1-r^2)/[r-(g+1)]}$	$\mathbf{t} = \frac{b_i}{se_{b_i}} \quad 0 \leq i \leq g$
	The polynomial is useful if $F > (F_\alpha(g, r-(g+1)))^*$	Coefficient b_i is useful if $\mathbf{t} < -(\mathbf{t}_\alpha(r-(g+1)))^*$ or $\mathbf{t} > (\mathbf{t}_\alpha(r-(g+1)))^*$

* F_α and $\mathbf{t}_\alpha$ are tabular values of the corresponding distribution with a level of significance α and degrees of freedom specified by their parameters

3.4 Selection Algorithm

In this algorithm t and ϵ are the allowed tolerance for the available processing time and the algorithm error. Additionally, n is the problem size, $T(n)$ is the efficiency function, $E(n)$ is the efficacy function, $f(I)$ is the solution algorithm expressed as a function that maps a problem instance I to a solution x, and $z(x)$ is the objective function (expression 1). Consequently with $f_E(I)$ the exact solution is found, and with $f_A(I)$ the approximate solution is obtained.

The first part of the selection algorithm is for problems in which suboptimal solutions are unacceptable. In this case it is necessary to check if the processing time predicted by $f_E(I)$ lies within the tolerance interval defined by t. In such a case the solution x of an instance I is found using an exact method, otherwise the algorithm ends without result.

The second part is for problems where suboptimal solutions are permitted. The first action is to verify if the exact method is adequate, else the heuristic method is evaluated. The exact method is adequate if the processing time predicted by $T_E(n)$ lies within the tolerance given by t, in this case the solution x is found using it. Otherwise, the heuristic method is adequate when the predicted processing time and the error lie within the tolerance intervals, in this case an initial solution x is obtained using the heuristic method, and from this one, new and possible better solutions are generated in y through successive runs. The number of runs is determined by dividing the tolerance time by the estimated processing time. Finally, if none is adequate, the algorithm ends without result. The algorithm previously described is the following:

Algorithm

```
Begin
    real t, ϵ ; // tolerances
    integer n; // problem size
    if only an optimal solution is acceptable then
       if T_E(n) ≤ t then
         x = f_E(I);
       else finish without solution
       end_if
    else
       if T_E(n) ≤ t then
         x = f_E(I);
       else
         if T_A(n) ≤ t and E_A(n) ≤ ϵ then
           if z(y) < z(x) then
             x = f_A(I);
             for i = 2 to t/T_A(n)
               y = f_A(I);
               if z(y) < z(x) then
                 x = y
               end_if
             end_for
           else finish without solution
           end_if
         end_if
       end_if
    end_i
End
```

4 Experimental Results

4.1 Results of Algorithm Behavior

In order to obtain the tabular description of the algorithm behavior, 40 experiments were conducted for each instance. 17 instances of wide size range and with known optimal solution were generated. These belong to the same class and were mathematically obtained using the Uncoupled Components Method [11].

Each test instance was solved using a Branch&Bound algorithm and the Threshold Accepting Algorithm. Tables 2 and 3 show a subset of the results of these tests. The second and third columns of Table 2 show the problem sizes for the test cases; while the last two columns show the performance results. Table 3 show the results obtained using the Threshold Accepting Algorithm. The difference percentage with respect to the optimal is shown on columns two through four. The last column shows the execution time of the algorithm.

Table 2. Exact solution using Branch&Bound

Instance	Sites	Queries	Optimal Value	Time(sec.)
I1	2	2	302	0.05
I2	18	18	2719	1.15
I3	20	20	3022	3.29
I4	32	32	* 4835	**
I5	64	64	* 9670	**
I6	128	128	* 19340	**
I7	256	256	* 38681	**
I8	512	512	* 77363	**

∗ Solution value determined using the Uncoupled Components Method
∗∗ Problem size exceeds algorithm implementation capacity

Table 3. Approximate solution using Threshold Accepting

Instance	% Difference (deviation from optimal)			Time (sec.)
	Best	Worst	Average	
I1	0	0	0	0.03
I2	0	141	10	0.3
I3	0	0	0	0.4
I4	0	78	4	1.2
I5	0	100	20	6.1
I6	0	140	36	43.6
I7	0	405	88	381.2
I8	66	383	215	3063.4

4.2 Performance Functions

The procedure for obtaining the performance functions was implemented with the mathematical package Matlab and the statistical software Origin.

Tables 4 and 5 show the best polynomial functions that were found using the regression method applied to the Threshold Accepting and the Branch&Bound algorithms.

Figure 1 shows the efficiency functions for both algorithms. A sixth degree polynomial was obtained for Branch&Bound, whereas a third degree polynomial was found for Threshold Accepting. Notice that for small instances the first outperforms the second, whereas for large instances the situation is just the opposite, and there is a crossing point between the two functions, which defines the dominance regions of the algorithms.

Due to the large spread of the efficacy results for the Threshold Accepting Algorithm, three polynomials were determined (Figure 2). For the best, average and worst cases the resulting polynomials were of first, third and first degrees.

Table 4. Polynomial functions for efficiency

Algorithm	**Polynomial Function $T(n)$**
Threshold Accepting	$-0.31458651 + 6.7247624E-5n + 4.3424044E-10n^2$ $-6.1504908E-17n^3$
Branch&Bound	$+0.0036190847 + 4.4856655E-4n - 4.4942872E-7n^2$ $+2.5914131E-10n^3 - 5.4339889E-14n^4$ $+2.5641303E-18n^5 + 2.4019059E-22n^6$

Table 5. Polynomial functions for efficacy (Threshold Accepting)

Test Case	**Polynomial Function $E(n)$**
Large instances (best case)	$-0.45443448 + 2.2522053E-5n$
Large instances (average case)	$+2.0494162 + 1.1588697E-4n - 1.8006418E-11n^2$ $+6.3173862E-19n^3$
Large instances (worst case)	$+56.62439 + 0.00011n$
Random problems (average case)	$-0.23663 + 0.00049n$

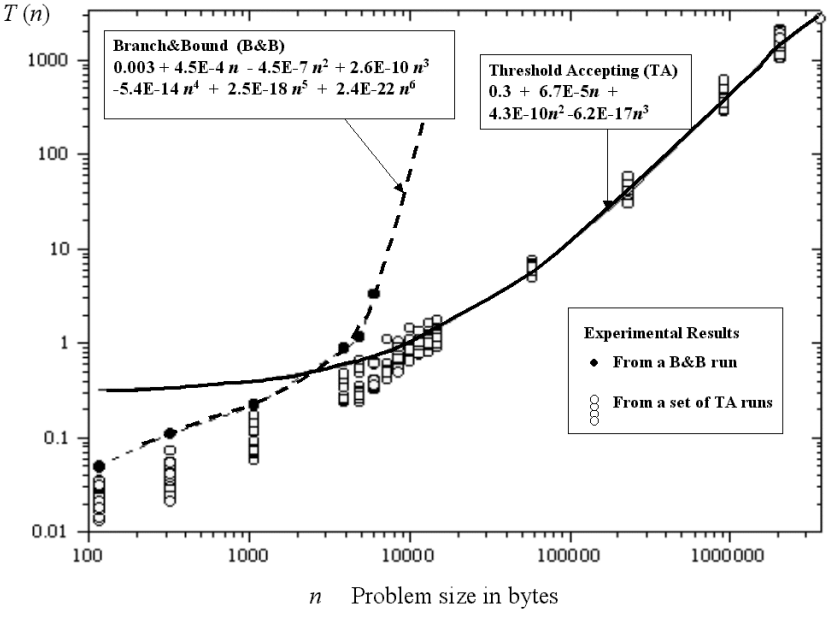


Fig. 1. Graph of the efficiency functions

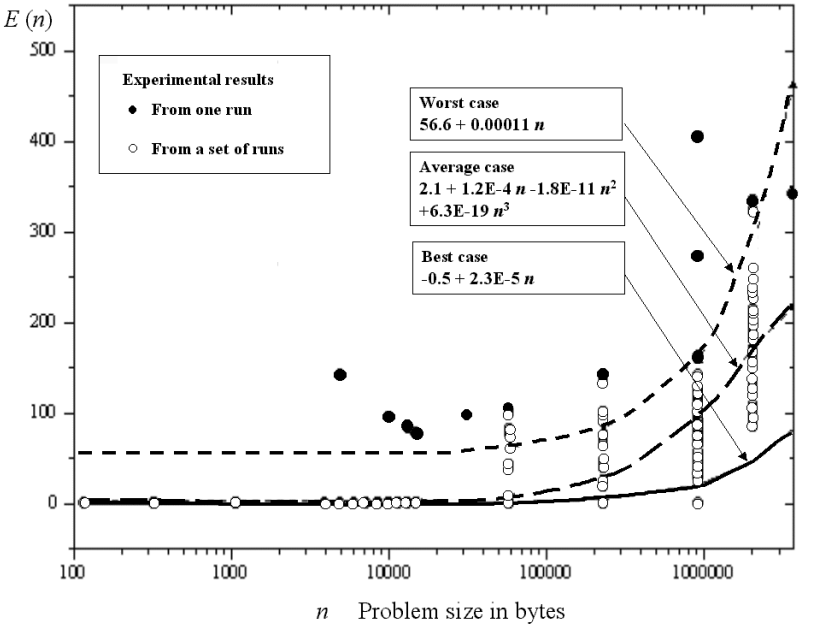


Fig. 2. Graph of the efficacy functions for the Threshold Accepting algorithm

5 Final Remarks

This paper shows that by finding the performance functions that characterize the exact and heuristic algorithms, it is possible to automatically determine the most adequate algorithm given the problem size. Also, the characterization helps us to better understand their behavior. For example, it defines regions in which one algorithm outperforms the other, as opposed to the traditional approaches, which oversimplify algorithm evaluation; i.e., they claim that one algorithm outperforms the other in all cases, which is not always true.

For demonstration purposes the performance functions for Branch&Bound and Simulated Annealing were obtained when applied to the solution of the database distribution problem modeled by DFAR. The experimental results have proved that Branch&Bound is satisfactory for small problems, Simulated Annealing is promising for large problems, and there exists a crossing point that divides both regions.

We are planning to integrate our work with another model, developed by us, to select the best between different heuristic algorithms.

References

1. Ahuja, R.K., Kumar, A., Jha, K.: Exact and Heuristic Algorithms for the Weapon Target Assignment Problem. Submitted to Operation Research (2003)
2. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. New Jersey, Prentice-Hall (1982)
3. Borghetti, B. J.: Inference Algorithm Performance and Selection Under Contrained Resources. MS Thesis, AFIT/GCS/ENG/96D-05 (1996)
4. Hooker, J.N.: Testing Heuristics: We Have it All Wrong. Journal of Heuristics (1996)
5. Hoos, H.H., Stutzle, T.: Systematic vs. Local Search for SAT. Journal of Automated Reasoning, Vol. 24 (2000) 421–481
6. Gent, I. P., MacIntyre, E., Prosser, P., Walsh, T.: The Scaling of Search Cost. Proceedings of AAAI'97. Mit Press (1997) 315–320
7. Pérez, J., Pazos, R.A., Romero, D., Santaolaya, R., Rodríguez, G., Sosa, V.: Adaptive and Scalable Allocation of Data-Objects in the Web. Lectures Notes in Computer Science, Vol. 2667. Springer-Verlag, Berlin Heidelberg New York (2003) 134–143
8. Pérez, J., Pazos, R.A., Romero, Cruz, L.: Análisis de Complejidad del Problema de la Fragmentación Vertical y Reubicación Dinámica en Bases de Datos Distribuidas. 7^{th} International Congress on Computer Science Research, México (2000) 63–70
9. Morales, L., Garduño, R., Romero, D.: The Multiple-minima Problem in Small Peptides Revisted, the Threshold Accepting Approach. Journal of Biomelecular Structure & Dynamics, Vol. 9, No. 5 (1992) 951–957
10. Pérez, J., Pazos, R.A., Vélez, L. Rodríguez, G.: Automatic Generation of Control Parameters for the Threshold Accepting Algorithm. Lectures Notes in Computer Science, Vol. 2313. Springer-Verlag, Berlin Heidelberg New York (2002) 119–127
11. Cruz, L.: Automatización del Diseño de la Fragmentación Vertical y Ubicación en Bases de Datos Distribuidas Usando Métodos Heurísticos y Exactos. M.S. thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey (1999)

Adaptive Texture Recognition in Image Sequences with Prediction through Features Interpolation

Sung Baik[1] and Ran Baik[2]

[1] Sejong University, Seoul 143-747 , KOREA
sbaik@sejong.ac.kr
[2] Honam University, Gwangju 506-090 , KOREA
baik@honam.ac.kr

Abstract. This paper presents a prediction method for an efficient on-line model modification in adaptive texture image recognition issue. The approach builds a close-loop interaction between object recognition and model modification systems. Object recognition applies an advanced RBF classifier in order to recognize objects on a current image of a sequence. Model modification manipulates the RBF classifier models by changing the structure of the classifier or/and parameters of classifier nodes in order to adapt to changing situations. For efficient model modification, this work includes modification through prediction that classifier models are changed in advance for adaptation if it is possible to predict the modification patterns. For experimentation, the change of texture characteristics has been investigated over a sequence of texture images acquired under dynamic perceptual conditions. Texture characteristics on images in a sequence have been extracted by Gabor spectral filtering, Laws' energy filtering and Wavelet Transformation filtering. The results of the investigation justify the need for an on-line model modification over the entire sequence of images in order to preserve the system recognition capability, and present the possibility of prediction by finding partial patterns of texture characteristics change. According to experimental results, it is seen that the prediction in model modification can enhance the competence of the system through experimentations.

1 Introduction

Adapting a visual system to time varying environments, an integration of computer vision processes with on-line learning/adaptation processes is required. This paper presents an on-line adaptation mechanism for a RBF classifier including a prediction function.

The approach builds an On-Line Model Adaptation process on top of the traditional Image Analysis process. Let us assume that texture-based image analysis is composed of (1) image processing for feature extraction, (2) image data classification, and (3) image segmentation into homogenous regions based on the classification results. In the traditional approach, off-line learned models are used for image data classification. Since texture characteristics can change away from a given model, model data have to be adjusted on-line to reflect ongoing change in texture characteristics. Therefore, the approach arranges a closed-loop mechanism for model

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 425–432, 2004.
© Springer-Verlag Berlin Heidelberg 2004

data adjustment according to the perceived changes. Reinforcement Generation module, Model Modification module and Strategy Selection module execute the perception of these changes and dynamic model manipulation.

2 Investigation of Texture Characteristics on Images in a Sequence

2.1 Texture Images in a Sequence

A sequence of images has been acquired by a camera with a b&w (black & white) option (240x320 pixels, each pixel in the 256-grayscale). Images were registered in a lab-controlled environment under (i) smooth changes in the distance between the camera and the scene, and (ii) varying lighting conditions. The distance was gradually decreased and the lighting source was gradually displaced across the scene. The next image was registered under the distance reduced by 4%. Total of 22 b&w images was obtained from the same scene. The scene contained four texture areas (class A, B, C, and D). Each texture area has both spectral and structural characteristics. Four selected images (image 1, 10, 11, and 22) were selected from a sequence to show that the resolutions of these images changed as the camera approached the scene gradually. Whereas there is similarity between two close images (image 10 and 11), there are significant changes between two distant images (image 1 and 10, or image 11 and 22) in Figure 1(a). There exist some noises on the texture images when images are perceived. In Figure 1(a), circles on class D of image 1 and class B of image 10 represent some noises occurred due to wrinkles of fabric. Figure 1(b) illustrates the change of image resolutions of the magnified patches over a sequence. A sequence of these patches means that some detailed and visible information is brought over the increasing resolution. Such a change corresponds to the change of overall texture feature values in the selected patches shown in Figure 1(c).

2.2 Texture Feature Extraction

Each incoming image is processed to extract texture features by using the most popular texture feature extraction methods, which are 1) Gabor spectral filtering [1], 2) Laws' energy filtering [2,3], and 3) Wavelet Transformation [4-7]. Those methods have been widely used by researchers and perform very well for various classification and image segmentation tasks. For efficient extraction of texture characteristics additional filtering steps are required [8]. The 7x7 averaging filter is applied to estimate local energy response of the filter. Next, non-linear filtering is applied to eliminate smoothing effect at the borderline between distinctive homogeneous areas. The non-linear filter computes standard deviation over five small windows spread around a given pixel. The mean for the lowest deviation window is returned as the output. Values of each texture feature are subject to a normalization process to eliminate negative imbalances in feature distribution.

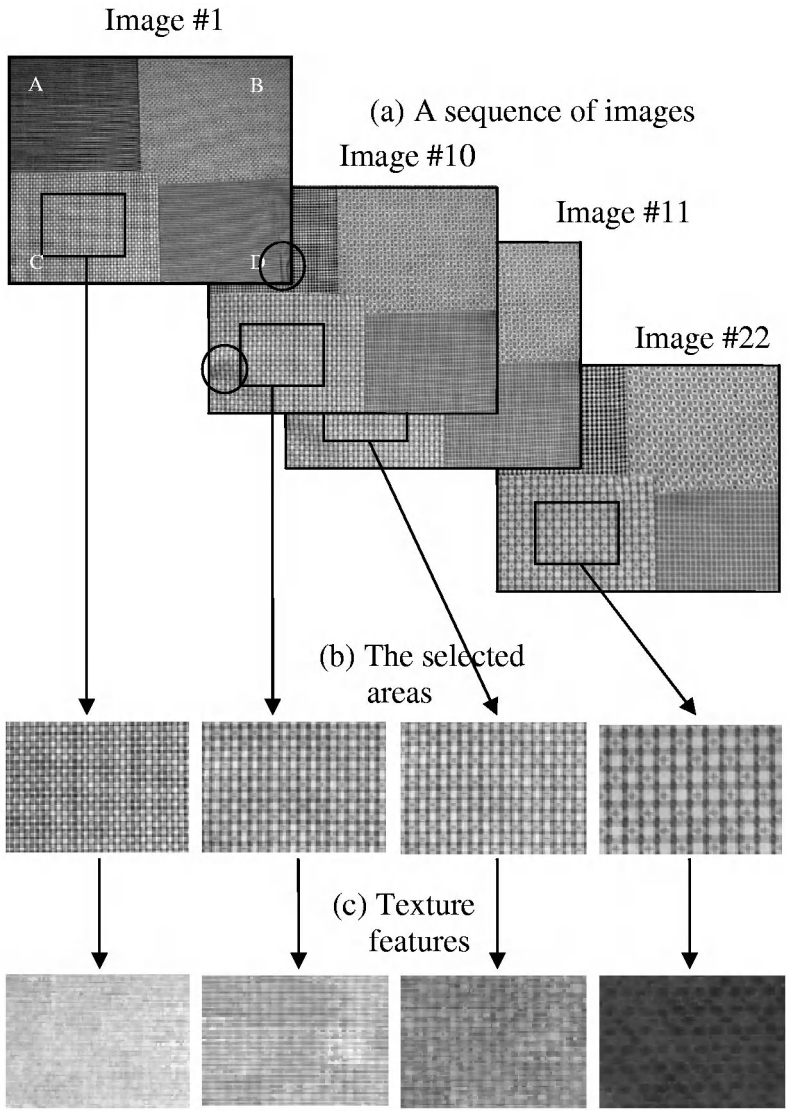


Fig. 1. Image sequence of the experimental domain (Total 22 images. A scene composed of four textured fabrics. The distance and lighting conditions gradually change from one image to another).

Gabor filters are useful to deal with the texture characterized by local frequency and orientation information. Gabor filters are obtained through a systematic mathematical approach. A Gabor function consists of a sinusoidal plane of particular frequency and orientation modulated by a two-dimensional Gaussian envelope. A two-dimensional Gabor filter is given by (1).

$$G(x,y) = \exp[\frac{1}{2}(\frac{x}{\sigma_x^2} + \frac{y}{\sigma_y^2})]\cos(\frac{2\pi x}{n_0} + \alpha) \quad (1)$$

By orienting the sinusoid at an angle α and changing the frequency n0, many Gabor filtering sets can be obtained. An example of a set of eight Gabor filters is decided with different parameter values (n0 = 2.82 and 5.66 pixels/cycle and orientations $\alpha = 0^\circ, 45^\circ, 90^\circ$, and 135°).

Laws' energy filters based on five dimensional vectors are used as an energy filter bank. It consists of 25 filters which can be derived from the weights of L5=[1,4,6,4,1], E5=[-1,-2,0,2,1], S5=[-1,0,2,0,-1], R5=[1,-4,6,-4,1], and W5=[-1,2,0,-2,1]. The respective specifications are Level, Edge, Spot, Ripple and Wave detection, in which convolving and transposing each other produce various square masks of 25 filters. Filters S3S3 and R5R5 respond to frequency. The rest of the filters are hand-crafted and have significant spatial geometric sensitivity. They are considered as special cases of Gabor filters.

Texture feature extraction algorithm based on wavelet transform provides a non redundant signal representation with accurate reconstruction capability, and forms a precise and uniform framework for the signal analysis at different scales [4]. In this work, 24 wavelet filters are generated by using three Daubechies wavelets (db1, db2, db3) and five biorthogonal wavelets (bior1.3, bior2.4, bior3.7, bior4.4, bior5.5) at one, two and three scale decomposition. There is no criterion to determine the decomposition level that yields the best discriminations. In practice, deeper level decompositions do not contain significant information.

2.3 Examples of Feature Sample Distribution

Figure 2 shows changes in feature distribution of class A and C over a sequence of images when one of gabor filters has been applied to extract texture features. Each point represents a mean value of a feature sample distribution in each class of an image. The mean values increase or decrease over a sequence of images. The mean value of the sequence sample distribution simply increases in class A. Between image 1 and 14, the mean feature value slowly increases and it rapidly increases in the rest range. Also, it is seen that over the first half of the sequence sample distribution is translated towards the higher feature values, and then reverses this translation trend towards the lower feature values in class C.

The change of feature mean values in each class over a sequence of images can be represented by a polynomial found by interpolating a mean value graph. The interpolation method is *least square curve fitting* [9] which is defined as follows:

Definition: Let $w_0, w_1, ..., w_m$ be a set of positive constants (weights). Given data points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, $m > n$, find $p^*(x) \in \mathrm{P}_n$, where P_n is the set of all polynomial of degrees n, such that $\sum_{i=0}^{m} w_i [p^*(x_i) - y_i]^2$ is minimized.

The curve graphs of these polynomials in Figure 2 are interpolation results in class A (P1) and C (P2). These two polynomials are as follows:

$$P1 = 0.000012x^6 + 0.00145x^5 + 0.05311x^4 - 0.79385x^3 + 5.3643x^2 - 1.4758x + 15.0481$$

$$P2 = -0.000037x^6 + 0.002369x^5 - 0.051568x^4 + 0.40033x^3 - 0.42512x^2 + 2.4604x + 5.8741$$

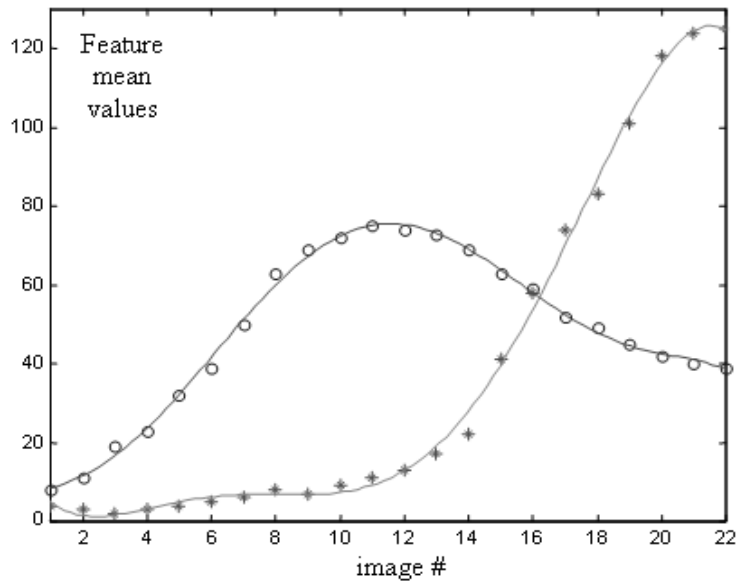


Fig. 2. The change of feature distribution over a sequence of images ('*' and 'o' marks indicates class A and C, respectively.)

3 Object Recognition and Model Modification

A RBF classifier [10-13], with Gaussian distribution as a basis, was chosen for texture data modeling and classification. This is a well-known classifier widely used in pattern recognition and well-suited for engineering applications. Its well-defined mathematical model allows for further modifications and on-line manipulation with its parameters and structure. The RBF classifier models a complex multi-modal data distribution through its decomposition into multiple independent Gaussians. Sample classification provides a class membership along with a confidence measure of the membership.

A feedback reinforcement mechanism [8] is designed to provide feedback information and control for the on-line adaptation of the classifier. This feedback exploits classification results on the next segmented image of a sequence.

Reinforcement parameters are analyzed in relation to the structure and parameters of the classifier. First, the system selects strategies (called behaviors) for the classifier modification. Second, it binds reinforcement data to the selected behaviors. Finally,

the behaviors are executed. There are four behaviors for the RBF classifier modification [10] that can be selected and executed independently: (1) *Accommodation*, (2) *Translation*, (3) *Generation*, and (4) *Extinction*. Each behavior is implemented separately using mathematical rules transposing reinforcement parameters onto actions of RBF modification.

Accommodation and Translation behaviors modify the classifier parameters only. This modification is performed over selected nodes of the net. The basis for Accommodation is to combine reinforcement parameters with the existing node parameters. The result of Accommodation is adjusted function spread. The node center does not change/shift through the feature space. The goal for Translation is to shift the node center in the direction of reinforcement without modifying the spread of the function. Combining Accommodation and Translation, the system can fully modify an existing node of the classifier.

Generation and Extinction behaviors modify the classifier structure by expanding or pruning the number of nodes. The basic idea of Generation is to create a new node. A node is generated when there is (1) a significant progressive shift in function location and/or (2) an increase in complexity of feature space, for example, caused by the increase in the multi-modality of data distribution. The goal of Extinction is to eliminate useless nodes from a classifier. Extinction is activated by the utilization of classifier nodes in the image classification process. Nodes, which constantly do not contribute to the classifier, are disposed. This allows for controlling the complexity of the classifier over time.

Additional *Prediction* behavior has been developed to progress the effects of accommodation and translation. Prediction magnifies the adjustments applied to the node boundary and node position in the feature space. This behavior is applied when there is a directional and persistent change in object characteristics.

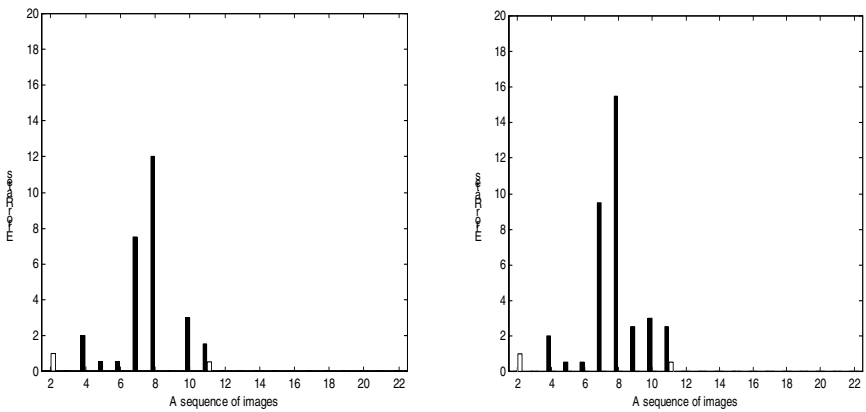


Fig. 3. Experimental results with the texture image sequence shown in Figure 1. Left and right diagram indicates classification errors before model modification and after model modification, respectively (Black bar: Class A and White bar: Class D).

4 Experimental Results

This section presents experimental results with the texture image sequence shown in Figure 1. Classification errors are registered for each new incoming image I(i+1) before the NN-RBF classifier is modified and after it is modified over the I(i+1) image. There are two types of error rates registered: 1) error rate with four behaviors of RBF classifier modification excluding prediction, and 2) error rate including prediction. Because the system goes through every image of a sequence, the modified classifier over the I(i+1) image is then applied to the next image. The results show a dramatic improvement in both types of error rates. Both error rates achieve almost zero level after the classifier is evolved over images of a sequence.

Figure 3 shows the classification error rates (before the NN-RBF classifier is modified) when including and excluding prediction, respectively. There are some improvement on image 7, 8, 9 and 11 since it was possible to predict some patterns of texture characteristics changes between image 7 and image 11 on the basis of patterns found from the range of the previous images.

Figure 4 shows two diagrams to present how to improve error rates between before modification and after modification on image 7 and 8, respectively. Whereas there is no difference in the number of evolution steps between including prediction and excluding prediction on image 7, modification including prediction is more easily achieved on image 8, on which classification error is the highest.

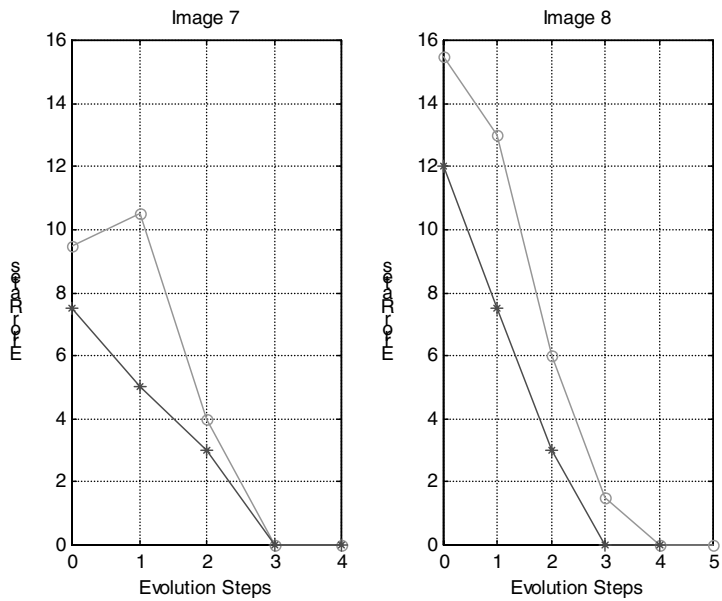


Fig. 4. The classification error rate change during model modification ('o' and '*' marks indicate model modification without prediction and with prediction, respectively.)

5 Conclusions

This paper provided an investigation of texture characteristics of images acquired under changing perceptual conditions. The results of the investigation justify the need for an on-line model modification over the entire sequence of images in order to preserve the system recognition capability, and present the possibility of prediction by finding partial patterns of texture characteristics change. The methodology developed has been tested on a variety of texture recognition problems in image sequences. The results demonstrated that on-line adaptation of the RBF classifier resulted in effective object classification over image sequences where object appearance was adversely affected by changing perceptual conditions such as resolution and lighting. According to experimental results, an addition of prediction to model modification enhanced the competence of the system through experimentations. Since prediction can reduce the evolution step in RBF classifier modification, the performance of the system was improved.

References

1. M. Farrokhnia and A. Jain. A multi-channel filtering approach to texture segmentation. Proceedings of IEEE Computer Vision and Pattern Recognition Conference, pp. 346–370, 1990.
2. M. Chantler, The effect of variation in illuminant direction on texture classification, Ph D Thesis, Dept. Computing and Electrical Engineering, Heriot-Watt University, 1994.
3. K. Laws. Textured image segmentation. Ph.D. Thesis. Dept. of Electrical Engineering, University of Southern California, Los Angeles, 1980.
4. M. Unser. Texture classification and segmentation using wavelet frames, IEEE Transactions on Image Processing. 4-11, 1549–1560, 1995.
5. S. Mallat. Multifrequency channel decompositions of images and wavelet models, IEEE Transactions on Acoustics, Speech and Signal Processing, 37-12, 2091–2110, 1989.
6. C. Chen. Filtering methods for texture discrimination, Pattern Recognition Letters, 20, 783–790, 1999.
7. T. Chang and C. Kuo. A wavelet transform approach to texture analysis, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 4, 661–664, 1992.
8. P.W. Pachowicz, A learning-based Semi-autonomous Evolution of Object Models for Adaptive Object Recognition, IEEE Trans. on Systems, Man, and Cybernetics, 24-8, pp. 1191–1207, 1994.
9. B. Carnahan, H.A. Luther, and J. O. Wilkes, Applied Numerical Method, John Wiley & Sons, INC
10. S. W. Baik, Model Modification Methodology for Adaptive Object Recognition in a Sequence of Images, Ph.D. Thesis, George Mason University, 1999.
11. M.T. Musavi, et al., A Probabilistic Model for Evaluation Of Neural Network Classifiers, Pattern Recognition, 25-10, pp. 1241–1251, 1992.
12. S. Chen, C.F.N. Cowan, and P.M. Grant, Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, IEEE Transactions on Neural Networks, 2-2, pp. 302–309, 1991
13. K. Wladyslaw and S. Pawel, Kernel Orthnormalization in Radial Basis Function Neural Networks, IEEE Transactions on Neural Networks, 8-5, pp. 1177–1183, 1997

Fuzzy Matching of User Profiles for a Banner Engine

Alfredo Milani, Chiara Morici, and Radoslaw Niewiadomski

Department of Mathematics and Computer Science
University of Perugia Italy
{milani,cmorici,radek}@dipmat.unipg.it

Abstract. Most advertisement systems widely used in Internet try to improve advertisement process by targeting specific groups of potential customers. Many systems exploit the information directly provided by the user and the data collected by monitoring user activities in order to built accurate user profiles, which determines the success of the advertisement process.
This paper presents a solution to the problem of targeting advertisement information when minimal knowledge about anonymous internet user is given. In particulary as, for example, in the case of search engines, the user remains anonymous and his interaction with the service can be very limited. In this case the information about him is sparse and based only on the keywords and the data submitted by the HTTP request. The proposed architecture is based on the use of predefined profiles and the computation of fuzzy similarities in order to match the observed user with appropriate target profiles. The notion of fuzzy similarity presented here is based on the theoretical framework of the Łukasiewicz structure, which guarantees the correctness of the approach.

Keywords: User Profiling, Fuzzy similarity, Soft Computing, Search Engine, Online Advertisement, E-commerce.

1 Introduction

It is a common fact that the dynamic growth of Internet investors is based on the expectation of incomes from advertisement in the global network. The most efficient way to capture user's attention is to provide him a personalized message,which can be generated by collecting in advance as much information as possible about him. In the case of many internet portals this problem seems to be "quite easy" to solve, because this kind of services usually requires some mechanism (like login procedure) to identify users. Systems based on authorization have the opportunity to collect systematically information about user either by using questionnaires or by tracing his choices, which permits a deep analysis of long- and short-term interests of the users. In the literature [8] is also described a method based on server log file analysis for session reconstruction in case when neither authentication procedure nor cookies are used. Another solution called

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 433–442, 2004.
© Springer-Verlag Berlin Heidelberg 2004

personal advertising profile [1] is based on registering user feedback for showed banners. Those methods are often denoted as individual filtering methods.

Another widely used technology for building personalized service is collaborative filtering. This requires to gather data about (anonymous) users transactions in order to find some of them who have similar behavior to the given one. In [2], i.e., it is presented a system which is based on multidimensional ranking of the content which is then the basis for user classification. Collaborative filtering could be supplemented by the use of some data mining techniques for pattern discovery like clustering [5] or discovery of association rules [6] and others.

Unfortunately there are many internet interactions which don't offer the possibility of solid observation of users behavior. In particulary typical search engine interaction is based on two steps request-response pages schema, and user would not have any reason to accept other longer ways to receive response (for example through login procedure). Once received the rewarding page he usually abandons the search engine site choosing a link, which he is interested of. The advertiser should catch him before he leaves, i.e. at the response page. In this scenario often only one HTTP-protocol request is given. We have called this situation *minimal knowledge hypothesis*.

This paper proposes an architecture in which the data are used immediately after they were submitted by an anonymous user to search engine and at the same time random advertisements generation is likely avoided.

2 User Profiling with Minimal Knowledge

In *minimal knowledge hypothesis* it is assumed that only the data from a user request page is available and we want to provide him a personalized advertisement information. For this purpose we use as bridge between advertiser and user a set of profiles defined for a particular problem. Profiles should correspond to some typical patterns of persons who could be interested on banner's topics. The general idea of the algorithm is to try to match, on the base of the available information, the actual user query to one user profile, chosen in a predefined set for the given problem. The match is realized by measuring the level of similarity degree. The second step of algorithm consists in choosing one banner for the profile just determined.

Profiles here play the role of individual users accounts from typical personalization models (i.e. [1]). However we first try to map a potentially infinite number of user requests to a finite set of profiles by modelling a cognitive process - the same which permits a human to describes intuitive likeness of two objects. Even if we use a word "profile" it is important to underscore that we don't try to guess user personality basing only on some keywords. Instead we try to classify some preferences using contingent data provided by the user query and general statistical data. We believe that even a user query to a search engine seems to provide very few information, this information can be amplified and interrelated in order to select appropriate user profiles.

The [3] contains an analysis of similar problem, but motivated by security reasons and privacy rights aspects. The work proposes the classification of all advertising methods for web services on some categories and the solution of

Adaptive Targeting. The *ADWIZ Advertisement System* is based only on keywords, does not use nor cookies nor other methods for storing the data. For any keyword is attributed the weight *display probability* of showing the banner. Next, weights are rearranged using performance data. Moreover the advertiser constraints are considered and the final solution is counted using variant Simplex method. The results of simulated experiment are compared then with random selection methods.

3 Architecture Overview

In order to model a similarity valuation user requests and user profiles have to be defined on the base of homogeneous criteria and data types. A relevant problem is that while the data from user request are usually "crisp" values, profiles are described using some sets, intervals or fuzzy sets. In order to exploit as much as possible the "few" available user data we put particular attention on users keywords, query's date and time and user IP number. It is important to point out that, for example in case of a search engine query, there is an unlimited number of user queries which can be constructed by using any words, moreover words can be misspelled and can belong to different languages.

3.1 Data from User Request

It is possible to obtain from HTTP request some useful information like the date and time of an action, preferable language of response or the IP number. We can also define (if needed) some properties depend on the data, like the boolean properties "foreigner" or "netscape-user". The HTTP requests submitted to the search engine contain also the user provided data - a set of key-words. All those variables possibly accompanied with others, based on values of HTTP's POST method parameters (like language or domain restriction in the case of popular search engines) will represent the input data to the profile matching system.

3.2 Data Representation in Profiles

In order to associate a user query to one of the n profiles it is necessary to express the properties of a user query in terms of intervals or sets. Lets suppose that user request is described by query's date/time and IP number. Instead of enumerate explicitly all possible time values of connection date/time constraints will be specified in the profiles by some fuzzy sets. In details date/time profile constraints are described in dual way: the day of the week is fixed when the time is represented by a set of trapezoidal time intervals associated with a day of the week. Those trapezoidal membership functions can be defined as:

$$tr(b,c,x)\begin{cases} x-b-\frac{1}{c-b} & a \le x \le b \\ \frac{1}{c-b} & b < x < c \\ c+\frac{1}{c-b}-x & c \le x \le d \\ 0 & x < a \quad or \quad x > d \end{cases}$$

$b,\ c \in [0,\ 24)$ and $a = b$ - $\frac{1}{c-b}$, $d = c + \frac{1}{c-b}$.

Also the IP number representation in profiles is not crisp. First we are not interested in IP number itself but the name of the country from which the connection is made. Having IP number it is easy to obtain this information using some databases. Using some statistical sources we can find a lot of particular and useful information which depends on location data. The decision about what properties are important is taken when defining profiles and it depends on the problem domain. That information is not directly inserted into profiles, because it could change and it can be updated in some dynamical way. Instead profiles are described by a set of countries.

Finally every profile can include in its definition a set of keywords of the same importance or with some weights assumed.

3.3 Banners

On the other side we have some banners. All of them are described by a set of keywords and set of weights w_{ij} - where w_{ij} express accuracy of banner B_j for profile P_i. We assume that every time the user makes query, one of m banners has to appear on response page. The keywords should be connected in some way with the subject of advertisement information. We can assume that both keywords and initial values of weights are defined by an advertiser (i.e. by ordering some profiles in "hierarchy of importance" instead of establish numeral values).

3.4 Ontologies

The most important information for proposed algorithm are usually user keywords. One of important elements of the solution is searching likenesses between key-words. Unfortunately neither single words nor their concatenation are quantitative values. Comparison between words can be made by using an ontology which allows to obtain more refined results than simple yes/no answers. Ontologies are necessary in order to classify and compare all those keywords according to their semantics. We suppose that the classification structure is a tree in which all keywords (leaves) are ordered in some categories and subcategories (nodes). Some keywords could be repeated as different leaves representing in this way two meanings of the keyword. The classification structure has a root node, which corresponds to a set of all possible classification paths. The classification tree represents a hierarchy in which more generic categories are closer to the root. We also assume that any keyword has at least a classification path and for every pair of keywords is possible to find a common path in the classification tree.

4 Fuzzy Similarity in Łukasiewicz Structure

The concept of similarity plays a leading role for our algorithm as we propose to use the computational model of this cognitive idea to express the fact that an object X *"is close to"* some set of objects of *similar nature*. The idea of similarity has found many mathematical descriptions. We briefly recall the basic concepts

of fuzzy similarity pointing out its relationship with Łukasiewicz structure. The detailed analysis of it can be found in [7].

The use of fuzzy similarity [9] is quite natural in order to evaluate and compare user profiles since it is a many-valued generalization of the classical notion of equivalence relation. More over fuzzy similarities and pseudo-metrics are dual concepts, as shown in [4].

Łukasiewicz structure is the only multi-valued structure in which the mean of many fuzzy similarities is still a fuzzy similarity. This property guarantees the correctness of the proposed algorithm for user profile comparison.

As proved in [7] fuzzy similarities can be used to compare pairs of objects. The various properties of objects can be expressed through membership functions f_i valued in [0,1]. The idea is to define *maximal fuzzy similarity* by computing the membership functions $f_i(x_1)$, $f_i(x_2)$ for comparing the similarity of objects x_1 and x_2 on each property i and then combining the similarity values for all properties.

Definition 1. Maximal fuzzy similarity. Since Łukasiewicz structure is chosen for membership of objects, we can define the maximal fuzzy similarity as follows:

$$S < x_1, x_2 >= \frac{1}{n} \sum_{i=1}^{n} (f_i(x_1) \leftrightarrow f_i(x_2)).$$

where x_1, $x_2 \in X$, f_i membership functions, $i \in \{1,...,n\}$ and $\leftrightarrow$ *(double residuum)* is the Łukasiewicz structure equivalence relation defined by:

$$a \leftrightarrow b = 1 - max\{a, b\} + min\{a, b\} = 1 - \mid a - b \mid$$

Non-zero weights can be associated to the different properties in order to express their different contribution to the similarity of the objects.

Definition 2. Weighted fuzzy similarity. A weighted fuzzy similarity can be defined by taking a weighted average on the single properties comparisons:

$$S(x_1, x_2) = \frac{\sum_{i=1}^{n} w_i (f_i(x_1) \leftrightarrow f_i(x_2))}{\sum_{i=1}^{n} w_i}$$

Note that this is still resulting a fuzzy similarity.

5 Algorithm for User Profile Matching

The most important part of the algorithm is to determine the most appropriate profile for a given user. The profile selection is made by evaluating similarities between the observed data and a set of possible profiles. The set of possible matching methods contain trivial cases of comparison between to boolean variables, matching crisp values with fuzzy values and finally two non-quantitative values (like two different words). More interesting is the case in which there are many or even infinite possible values for the variable.

We present now the computation model for establishing the level of similarity between them. Lets consider three different data types which describe user profiles for search engine advertisement: keywords, connection date/time, IP/user location. Finally the similarities values previously computed are finally composed to obtain the final result; the composition is still a fuzzy similarity [7].

5.1 Evaluating Keywords Similarity

The similarity between the set of *observed* keywords and the set of *target* keywords is based on evaluating similarity between pairs of keywords. We assume that a set $F=f_1,...,f_n$ of classification functions are available, each function f_j is such that given a keyword K_i, $f_j(K_i)=v_i$, returns v_i, the path from node K_i to the root in the classification tree represented by f_j.

A path is an ordered sequence of nodes $v_i=(n_0,...,n_k)$ where n_0 is the root and n_k is K_i, a path in the classification tree represents a set of categories/subcategories which define a particular meaning of a keyword. By extension $F(K_i)=v_i$ returns the set of classification paths for keyword K_i.

In addition we define:

- L the longest path from any leaf to root in the classification tree represented by F,
- l_{v_i} denotes the length k of path $v_i=(n_0,...,n_k)$,
- $l_{v_{ij}}$ denotes the length of the common path (i.e. the number of common arcs).

Moreover we admit that every keyword K_i and every v_i of K_i can have associated some weights w_{K_i} and w_{v_i} which express the importance of given keyword in the definition of profile and "the importance" of the certain meaning of keyword. Those values can depend for example from user origin (in case in which some key-words have two different meanings in two different languages). Only for the legibility and without losing the completeness of the solution those weights are omitted in formulas presented below.

Path similarity. The path similarity between $v_i,\ v_j$ is defined as:

$$S_p(v_i, v_j) = \frac{1}{2L}(2L - d(v_i, v_j))$$

where $d(v_i,v_j)$ can be seen as a "dissimilarity":

$$d(v_i, v_j) = (l_{v_i} - l_{v_{ij}}) + (lv_j - l_{v_{ij}}) = lv_i + lv_j - 2l_{v_{ij}}$$

$S_p(v_i,v_j)$ is a similarity since we can prove that $d(v_i,v_j)$ is a pseudo-metric.

Keywords Pair Similarity. Since every keyword is classified by a set of classification paths, the similarity $S_K(K_i,K_j)$ between two keywords K_i and K_j is defined as the maximum of $S_p(v_i,v_j)$ over each $v_i \in F(K_i)$ and $v_j \in F(K_j)$:

$$S_K(K_i, K_j) = max S_p(v_i, v_j)$$

Keywords Set Similarity. Let U be the set of keywords *observed* in the user query $U=(K_1,..,K_m)$ and every profile P_i is described by a *target* set of keywords $W_i=(K_1,..,K_n)$. Then a straightforward solution for determining the best matching set W_i for U is to consider the mean value of $S_K(K_i,K_j)$ over every pair of keywords of $K_i \in W_i$ and $K_j \in U$.

The adoption of a mean value reflects the intention that all keywords contribute to represent the *meaning* of a profile, on the other hand redundant or similar keywords do not contribute to increase the mean.

5.2 Evaluating Connection Time Similarity

Lets consider now the case of matching based on date/time of connection. This information has to be locate in every HTTP communicate. The similarity is made here between crisp value from the user query and the date/time constraints represented by a set of trapezoidal time intervals of every profile.

Given the user's date/time x_u and given, for every profile P_i, the date/time intervals in the profiles $[b_{ik}, c_{ik}]$ *(where* $1 \leq k \leq m$*, m - number of intervals of* P_i*)* the *date/time similarity* degree is computed into two steps:
Step 1. The greatest value of $tr(b_{ik},c_{ik},x_u)$ is computed (tr_{x_u}) for the user observed time x_u along all date/time intervals in all profiles:

$$tr_{x_u} = \max_{ik} tr(b_{ik}, c_{ik}, x_u),$$

Let b and c define the interval such that $tr(b,c,x_u)=tr_{x_u}$.
Step 2. The second step consists in computing the similarity degree between the membership of x_u for each time interval $[b_{ik}, c_{ik}]$ of each profile P_i and the best membership of x_u in all intervals, i.e tr_{x_u}. This similarity degree is computed by *maximal fuzzy similarity* as in DEF. 7 when $n=1$.

$$S_T(tr(b_{ik},c_{ik},x_u),tr_{x_u})=tr(b_{ik},c_{ik},x_u)\leftrightarrow tr(b,c,x_u)$$

5.3 Evaluating User Location

The information about country of user location can be easily obtained from the IP address. At the same time profiles are described by a set of countries. "Countries comparison" is possible only by establishing first some quantitative criterion, so additional information and properties are associated to each country, properties like annual income, population, religion etc. The similarity between countries is based on the similarity of the quantitative properties associated with them. In order to make a consistent comparison among values of properties in the different countries an ordering is induced by appropriate fuzzy sets; n fuzzy sets μ_j, associated to the n relevant properties for countries, are defined for each country

$$\mu_j(C) = \frac{p_j(C) - \underline{p}_j}{\overline{p}_j - \underline{p}_j},$$

where $p_j(C)$ returns the value of property j for country C, *j=1...n*, *n* number of properties, and:

$$\underline{p}_j = \min_C p_j(C), \quad \overline{p}_j = \max_C p_j(C).$$

The memberships $\mu_j(C)$ of the different properties are then compared and weighted using *weighted fuzzy similarity* (DEF. 2) thus obtaining the global degree of similarity between countries for profile P_i:

$$S_C(C_1, C_2) = \frac{1}{\sum_{j=1}^{n} w_{ij}} \sum_{j=1}^{n} w_{ij}(\mu_j(C_1) \leftrightarrow \mu_j(C_2))$$

where C_1, C_2 denotes the countries to compare and n is the number of considered properties. Weights w_{ij} are defined for every profile in order to point out the relevance of a property in the context of a profile.

5.4 Combining Similarities

After that n similarity values has been independently evaluated for different types of data, it is finally possible to combine them in order to find the target profile which best matches the current user.

Let m_{ij} be the value of similarity j for observed data and a profile P_i, then a decision function can be defined:

$$up_i = \frac{1}{\sum_{j=1}^{n} w'_{ij}} (\sum_{j=1}^{n} w'_{ij} m_{ij})$$

Again w'_{ij} are weights defined for every profile, they allow to express the relevance of the type of observed data for determining a certain profile.

Finally, the profile most similar to the user observed data can be determined by considering maximum value *(up)* of up_i.

Generic profile. The last improvement to the mechanism considered so far is the use of an generic profile if system finds difficulty to choose the profile for a user. Not recognized user (or not recognized enough) will correspond to the situation in which *up* value is very small. ($up < upvs$, where *upvs* is a fixed constant). It means that user's data doesn't match any particular profile definition. The generic profile accounts for this kind of situation ensuring the completeness of the algorithm.

6 User Profiling for Advertising on Search Engine

A typical application of our *minimal knowledge hypothesis* can be an advertisement campaign based on search engines. Lets the user query for matching algorithm be based on variables:

- user request string constructed from words and some operators,
- query date and time,
- user country.

Note that those three variables give an enormous set of possible user queries (especially considering all possible sequences of keywords) On the other side we have (usually quite small) set of m banners. As we said before in SECTION [2] we use a set of n profiles as middle tier between advertiser and user of search engine, (with $m \neq n$ in generally). Now we can define the problem as finding the profile which fits in the best way a user request and then matching the right banner with the chosen profile. The first part can be resolved using algorithm presented in [5.1] - [5.3] with $n=3$ in maximum weighted similarity formula (DEF. 2).

6.1 Ontology

In order to have an ontology for keywords classification, one can use search engine itself; because those services are able to categorize every user query. We have used the project DMOZ *Open Directory Project* [10] which is the largest categories directory and search engine edited by human. An ontology is freely accessible both in RDF format and as on-line search engine. Primarily it's designed as a web directory but we propose to use it as hierarchical structure which classifies words by assigning them available classification paths. Unfortunately it's not ideal for our purpose, because used classification not always consider some important semantic aspects. It is considered in future extensions the composition of various ontologies to obtain more suitable results.

6.2 Choosing Banners

On the other hand we have a set of banners. Every of them is described by a set of keywords and a set of weights w_{ij} - where w_{ij} express accuracy of banner B_j for profile P_i.

Once knowing the profile, the choice of the banner is made by looking for a maximum value of some formula along banner weights w_{ij}. At the beginning weights are predefined. After that they have to be modified depending on user (as a profile representative) interest of advertising information (feedback effect). Moreover the banner choice is parameterized by some random value and "randomization" decreases with the number of computation for B_j. In this way we avoid situation in which user receives every time the same banner and so the correctness of the algorithm is assured.

Step zero matching. Moreover, in the *step zero* algorithm tries to make direct matching between keywords of banners and request avoiding middle-tier of profiles. It would correspond to the situation in which user's keywords "are very close to" the description of banner. For this purpose is used the same ontology and the algorithm check all possible pairs of keywords (user's keyword, keywords of banner) looking for some with distance between each other less or equal to d.

7 Conclusion

An algorithm for fuzzy matching of user profile in order to personalize advertisement information has been presented. An original contribution of the presented work is a technique to amplify the minimal knowledge provided by the user, and to exploit it in the comparison process, which is based on a formally grounded fuzzy notion of *likeness*.

The proposed algorithm is a part of a more general architecture for user profile matching. Future work will consider to expand the system by the use of hierarchical relation between profiles. Profiles with different level of accuracy can be considered in a different way. The system can choose the profile more detailed if the decision can be made without much risk. We have just mentioned this problem by creating generic profile for not recognized users.

It is also planned to compose different ontologies and to introduce automated techniques for supporting the definition of profiles which are currently set manually. Automatic generation of profiles and weights can involve data-mining techniques.

References

1. Patrick Baudisch, Dirk Leopold, User-configurable advertising profiles applied to Web page banners, In Proceedings of the First Berlin Internet Economics Workshop, Berlin, 1997.
2. R. Burke: Semantic ratings and heuristic similarity for collaborative filtering, AAAI Workshop on Knowledge-Based Electronic Markets, AAAI, 2000.
3. M. Langheinrich, A. Nakamura, N. Abe, T. Kamba, Y. Koseki: Unintrusive Customization Techniques for Web Advertising, Computer Networks 31(11-16): 1259–1272 (1999).
4. P. Luukka, K. Saastamoinen, V. Könönen, E. Turunen: A classifier based on maximal fuzzy similarity in generalised Łukasiewicz structure, FUZZ-IEEE 2001, Melbourne, Australia.
5. B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Y. Sun, J. Wiltshire: Discovery of Aggregate Usage Profiles for Web Personalization, Data Mining and Knowledge Discovery 9, 2002.
6. B. Mobasher, H. Dai, T. Luo, M. Nakagawa: Effective Personalization Based on Association Rule Discovery from Web Usage Data, WIDM 2001, Atlanta, 2001.
7. C. Morici, R. Niewiadomski: A framework for a personalized advertising on Web based on maximal fuzzy similarity in Łukasiewicz structure, in Proceedings of IPMU 2004.
8. M. Spiliopoulou, B. Mobasher, B. Berent, M. Nakagawa: A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis, INFORMS Journal on Computings 15.
9. E. Turunen: Mathematics behind Fuzzy Logic. Advances in Soft Computing, Physica-Verlag, Heidelberg, 1999.
10. Web resources for Open Directory Project: *http://dmoz.org/about.html*.

Genome Database Integration

Andrew Robinson and Wenny Rahayu

Dept. Computer Science and Computer Engineering,
LaTrobe University, Australia
{wenny,aj5robin}@cs.latrobe.edu.au

Abstract. This paper presents a solution to many of the problems in Genome Database Integration including an integrated interface for accessing all genome databases simultaneously and the problem of a common interchange data format. The solution is the addition of a middle or mediation layer of a three layer approach. The solution provides a simple step by step approach to connect other existing genome databases quickly and efficiently. The internal data format used is a commonly used bioinformatics format called BSML, a subset of the XML standard. The architecture also allows easy addition and deletion of functionality. Finally an implementation of this solution is presented with the required support functionality to validate the proposed integration method.

1 Introduction

With introduction of mass genome sequencing machines and projects comes a large volume of sequence and annotation data. Much of the work that scientists in the area of bioinformatics do involves the use of all and parts of the data produced from these sequencing projects. In particular the genomic data needs to be in a format that is easily searchable by the annotation and actual sequence data for similarities which could infer new knowledge.

Over the years that genome data has been gathered many databases have been formed to attempt to make this data accessible and complete. However every time a new database is created to store the latest group of sequences the creators generally make a new format. This leads to making the databases as a whole not searchable and useable. Combining all the Genome data that is released in public domain is not a viable solution because of the huge demands that are required on the database storage-wise, performance-wise and maintenance-wise. For that reason there is a need to have many separate databases that can co-exist with a common method to interact with each other. It is expected that the integration of external databases should not affect the way in which the original databases are structured or interfaced to.

Section 2 in this paper provides an overview of related work in the area of Genome Database integration. Section 3 defines a new architecture for solving the problems. Section 4 describes the process of mapping external data to a common format. Section 5 discusses the implementation of the system architecture and mapping process.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 443–453, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Background

There are currently three basic types of solutions to the database integration problem. These are differentiated by the place of storage of data or the way data is referenced. The three types are 'hyper-linked', 'absorb locally' and 'query external' databases. In this section the later two types will be discussed as those concepts are used in this paper. Many of the example databases found have a composite of integration types.

While the 'hyper-linked' integration technique is widely used it is not appropriate for the solution that is proposed in this paper because it doesn't directly support mass integration. The technique gives, on all output, hyper text links to similar or related data in other databases which otherwise would be totally separate.

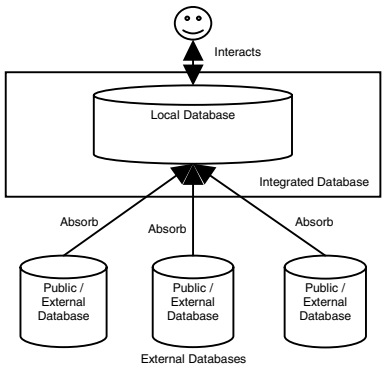


Fig. 1. The basic database structure for the 'absorb locally and query' style databases.

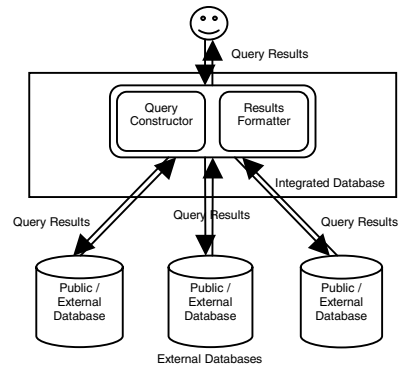


Fig. 2. The database structure for the 'external query' style databases.

Fig. 1 shows the general database structure of the 'absorb locally' style of databases. The data from the public or external databases are formatted and absorbed into the local database. The user then interacts with the local database. The problems faced with this style are usually to do with local data format and the importing/updating. Some benefits of these are that doing searches is faster providing a high performance computer is available. It also reduces the internet bandwidth used if there are a lot of searches being completed. Some examples of this type of database are a Protein finger-prints database called PRINTS-S [2] (extended version of original PRINTS), BioMolQuest [3] and GIMS [4]. The majority of Local Laboratory Information Management System (LIMS) databases also use this method to search their 'in-house' sequences against [5].

Fig. 2 shows the basic system architecture of the 'External query' style of database. The main distinguishing factor of this style is the fact that no data is actually stored long term on the local system. The 'Query Constructor' module takes a query from the user, converts it and runs in on each external / public database. The results are returned to the results formatter which formats them for the user.

Some examples, (mostly research papers,) are 'A Genome Databases Framework' referenced in [6], TAMBIS [7], and the database system specified in [8]. The main limitations are that most of the current implementations are very complex and some are hardware dependent.

3 Proposed Architectural Framework

In order to solve the problem of integration of Genome databases there needs to be several things available. These are:

1. An architecture that provides the easy addition of databases and formatting / manipulation modules. There is also a need for these extendibility features to be easily understood by the target users, researchers in bioinformatics, so they can be involved in its development.
2. A common 'language' that supports the interchange of data.
3. A platform that uses existing standard or common practices in data format. This really re-enforces the need for the first two points by the means that the user is already used to using the standard before which, (1) reduces the perceived complexity and learning curve of the architecture, and (2) with the use of standards brings previous knowledge and tools which are optimised for the situation.

The frame work structure chosen was a mediator to address the requirements of the first point noted above. A mediator is a collection of functionality that is designed to have functionality added and removed with minimal changes to layers above and below [9]. While mediators haven't been used yet in Genome database integration they have been proved extensively in other domains such as that presented in [9].

Bioinformatic Sequence Markup Language (BSML), a subset of XML, is used as the common data format for data integration because it is fairly source independent and already has support from some of the major public databases [10]. The use of BSML (or even XML) inherently solves the third problem mentioned above of standard data format for bioinformatics.

The proposed mediator is intended to be placed between the 'user interface' and the 'actual databases'. Internal to the mediator there is also three layers; these layers are the 'Service' layer, the 'Task' Layer and the 'Database' Layer. The architecture is depicted in fig. 3 which shows the main interactions and data/message flow between the main parts of the mediator. Fig. 4 shows the database modules in more detail. The face shape at the top of the figure represents the user that uses the mediator. Objects in these figures are represented by the rounded boxes and data/message flows are shown by arrows.

There are three 'Manager' objects in the architecture, one for each layer. The role of these is to assist in the independent extension of these layers and to assist the communication between the layers. Between each of these layers is a strict XML interface. The interface is defined in such a way that the functionality of the mediator can be extended independently.

The Services are responsible for coordination and advertising the work of the one or more tasks. Tasks are responsible for performing one operation on a given set of data which is obtained from the database classes after the format conversion.

There are two types of database objects; XML format and non-XML formatted. XML format consists of the databases that are already in a XML format. XML formatted are separated from the rest because they cane be reformatted by XSLT's.

As seen in fig. 3, the Database layer includes two types of caches; the 'Transform Cache' and the 'User Cache'. The 'Transform Cache' is used to store the latest transformed version of the source sequences. This allows the Mediator to retrieve quickly the sequences that are converted more often. The 'User Cache' is used for allowing the user to manually create 'save points' when manipulating sequences to

again further improve the resource use and speed of the mediator. The difference between the local and remote database layers is that the remote is actual pre-existing databases and the local is for data created / modified by the mediator.

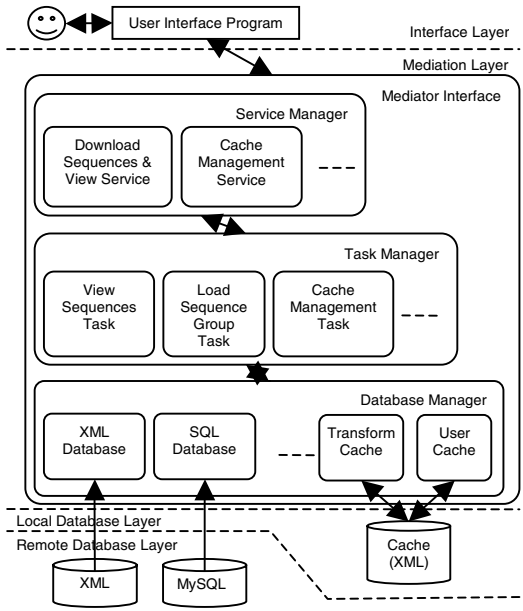


Fig. 3. High-level view of the Mediator

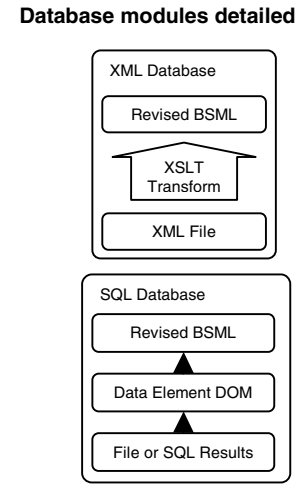


Fig. 4. Detailed view of the Database Layer

4 Genome Data Mapping Process

Up to date, one part of the mediator's description has been left out, that is, the process to map data from the external databases to the internal 'revised BSML' format. Section 4 will now describe the process of adding a new database to the architecture as well as describing how the actual mapping occurs. These descriptions will be done using examples taken from Swiss-Prot.

4.1 Revised BSML

While BSML was selected because it matched the need, there are still a few minor problems with the organisation of the data and naming standards. For these reasons a revised version of BSML is used as the mediators 'common data format'.

The motivation behind revising the BSML structure and format is (1) to make it follow the standards for 'good XML' (W3C Standards and recommendations [11]) (2) to make it more independent of the biological source it came from and (3) to separate the three distinct groupings of annotation. The three distinct groups are annotation of

the whole sequence, references to papers / other databases, and annotation about particular parts of the sequence.

The first step taken to improve the BSML format was to convert all the attributes to elements where needed. Fig. 5 below shows a snippet from the BSML format under the 'original' heading and the revision under the 'Revised BSML' heading.

Original:

```
<Sequence     id="G:186439"     title="HUMINSR"     molecule="rna"     length="4723"
comment="Human insulin receptor mRNA, complete cds.">
        …
</Sequence>
```

Revised BSML:

```
<sequence id="G:186429">
        <id>G:186439</id>
        <title>HUMINSR</title>
        <molecule>rna</molecule>
        <length>4723</length>
        <comment>Human insulin receptor mRNA, complete cds.</comment>
        …
</sequence>
```

Fig. 5. Convert BSML's data attributes to elements

The other revisions made are as follows:

1. Replace the 'Attribute' element with a structured set of elements so its more human readable and (2) it can represent more complex XML structures.
2. Add a series of new elements to allow the BSML format to represent a wider range of annotation. E.g. Keywords, dates, descriptions etc.
3. Separate the three distinct groups of sequence annotation.
4. The last revision is to rename some of the elements so that they are more intuitive to a human reader and follow a general standard.

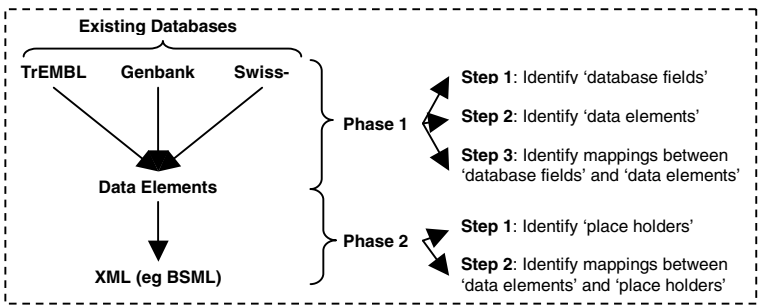


Fig. 6. Biological data mapping process

4.2 Two Phase Mapping Process

The mapping process consists of two major phases. The first is to map the individual fields in the source file (or database) to the data elements described below. The second is the mapping of the data elements to the revised BSML. The Data Elements middle point has been added to allow a more consistent mapping of multiple sources in XML and to separate the common parts of the mapping.

Fig. 6 below show the basic steps involved in the mapping process. Three public databases are shown at the top. The final output (an XML formatted document) is shown at the bottom. Data Transformations are represented by a unidirectional arrow.

4.2.1 Phase 1: File/Database to Data Elements Mapping

The first phase in the mapping process for files is transforming the external flat file format or relational database format into the data elements format. The data elements middle stage has been added to provide a method to get the data field names in a standard format to make searching on combined data more consistent. The examples in this section will be mapping a Swiss-Prot flat file to data elements.

4.2.1.1 Step 1: Identify the Source File/Database Fields

This step involves taking a 'full' source file or schema and marking each of the important fields. Fig. 7 shows each of the separate fields marked by a rectangle on a Swiss-Prot reference to a sequence.

```
RN [1]
RP SEQUENCE FROM N.A.
RC STRAIN=ATCC 35210 / B31;
RX MEDLINE=98065943; PubMed=9403685;
RA Fraser C.M., Casjens S., Huang W.M., Sutton G.G., Clayton R.A.,
```

Fig. 7. Example of identifying the fields from part of a Swiss-Prot flat file

4.2.1.2 Step 2: Identify the Data Element Fields

As this part is common between all database conversions of this type this part is done already. All that is needed is for the person creating a new database interface to understand each of the data elements so that they map the correct data to each. To assist the understanding of each field a brief description is given in the following figure.

Data Element Descriptions

This section gives a few examples of 'Data Element's. There are three examples shown here. These are the Sequence Data, Sequence Length and Sequence ID. The remaining Data Element Descriptions can be found in [1]. The numbers in the brackets are used for identifying the Data Elements in the latter mapping step.

Sequence Data [1] The sequence data 'Data Element' is probably the most important one of all since without it there would be no use for the rest of the Data Elements. It contains the actual Protein or DNA sequence bases.	**Sequence Length[2]** The Sequence Length Data Element stores the number of letters in the actual sequence data **Sequence Internal ID [3]** This is the unique ID from the source database. It is stored to all the updating of data in the future.

Fig. 8. Sample of the Data Descriptions

4.2.1.3 Step 3: Mapping Database/File Fields to 'Data Elements'

The process of mapping database/file Fields to 'Data Elements' consists of looking at each database/file fields and finding the match for that field in the Data Elements. The match can be identified by reading the descriptions of the fields presented earlier. If no match for a database/file field can be found then the field can be mapped to the Attribute data element in the section that the data comes from (i.e. the reference, feature or whole sequence).

Fig. 9 is an example of mapping a flat file database to the 'data elements'. The flat file is sourced from the Swiss-Prot database. This example is cut-down so it shows all the important features. A full mapping can be found in [1].

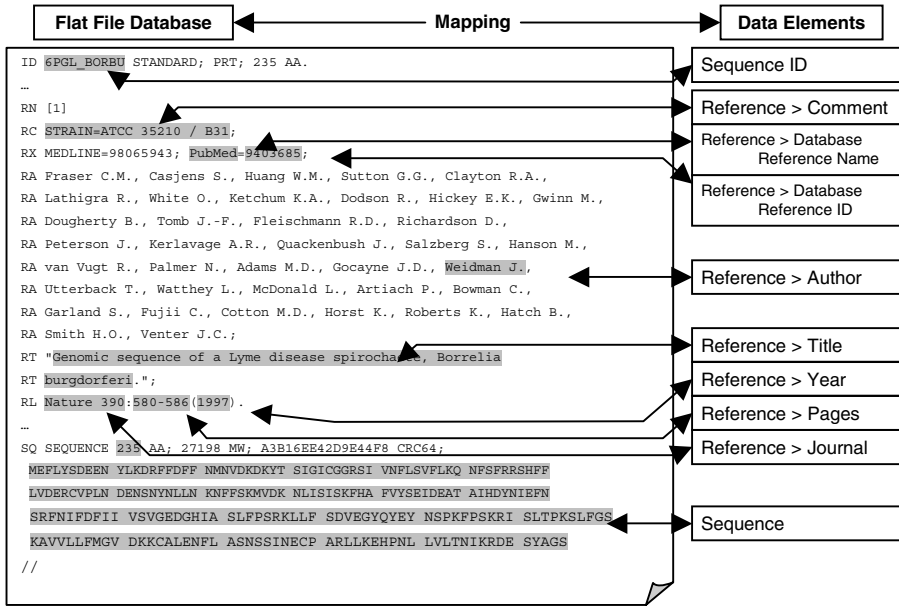


Fig. 9. Mapping between Swiss-Prot file and Data Elements

4.2.2 Phase 2: Data Elements to XML (Revised BSML) Mapping

The second phase of the mapping process consists of mapping the 'data elements' discussed in the previous section to the BSML format. As with some parts to the last conversion this phase of the mapping process is common to all.

4.2.2.1 Step 1: Identify the 'Place Holders' in Revised BSML

This step is done by taking a full example of a revised BSML document and locating all the places that could have data in them. Fig. 10 has and example of a partial revised BSML document in it. The 'place holders' are marked with '[px]', where x is replaced with a sequence number (eg. 1, 2, 3 etc.).

```
<?xml version="1.0"?>
<!doctype bsmlrev system "bsmlrevised_1_0.dtd">
<bsml>
    <definitions>
       <sequences>
          <sequence id="[p1]">
             <id>[p2]</id>
             <length>[p3]</length>
             <comment>[p4]</comment>
             …
             <keywords>
                <keyword>[p9]</keyword>
             </keywords>
             …
             <dbReferences>
                <dbReference>
                   <databaseName>[p14]</databaseName>
                   <databaseID>[p15]</databaseID>
                </dbReference>
             </dbReferences>
             <reference_tables>
                …
             </reference_tables>
             …
          </sequence>
       </sequences>
```

Fig. 10. Place holders marked in a revised partial BSML document

4.2.2.2 Step 2: The Mapping between 'Place Holders' and 'Data Elements'

The second step is mapping the 'place holders' found in step one to the 'data elements described in the previous phase. Fig. 11 shows a table mapping some 'data elements' to 'place holders'. Since the relationship between them can be many-to-many there can be multiple 'data elements' or 'place holders' in each column.

Data Element(s)	Place Holder
[3] Sequence internal ID	[p1] <sequence id="[p1]">
[3] Sequence internal ID	[p2] <id>[p2]</id>
[2] Sequence Length	[p3] <length>[p3]</length>
[5] Comments	[p4] <comment>[p4]</comment>
[10] Module	[p5] <molecule>[p5]</molecule>
[11] Version	[p6] <version>[p6]</version>
[4] Organism species	[p7] <organismSpecies>[p7]</organismSpecies>
[7] Organism classification	[p8]<organismClassification>[p8]</organismClassification>
[6] Keywords	[p9] <keyword>[p9]</keyword>
…	
[42] Feature Attribute value	[p33] <value>[p33]</value>
[40] Database Reference Database Name	[p34] <name>[p34]</name>
[39] Database Reference Database ID	[p35] <id>[p35]</id>
[12] Sequence Attribute Name	[p36] <name>[p36]</name>
[13] Sequence Attribute ID	[p37] <value>[p37]</value>
[1] Sequence Data	[p38] <sequence_data>[p38]</sequence_data>

Fig. 11. Mappings from data elements to place holders

5 Implementation

This section is to describe the actual implementation and the implementation decisions that have been made along the way.

The system has been implemented in Java Sevlets running in Apache Tomcat. The choice of Java for the implementation language was because of java's platform independence. Secondly Java has a really abundant set of functions for dealing with XML. Another less important reason for choosing Java is that Java is one of the most widely used languages in Bioinformatics.

5.1 The Basic User Interface

The user interface is outside the boundary of this project but is needed to show that the mediator works. The 'Interface Program' simply requests the available services from the 'Mediator Interface' and allows the user to select from the list (see fig 12). Upon selecting the 'Service', the Interface requests the settings from the 'Service' from which it constructs a series of text boxes to input the options (see fig 13). The user inputs the options and clicks 'Submit Query'. The Interface takes the options and executes the service. After which it displays the results (see fig 14).

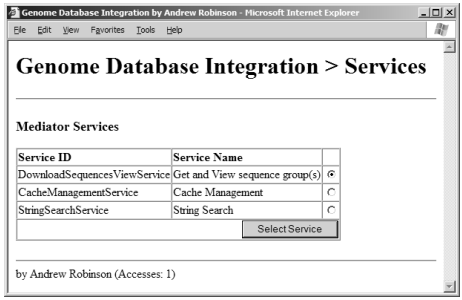


Fig. 12. Select a service on the 'Interface Program'

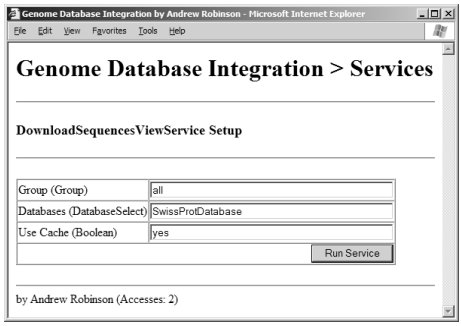


Fig. 13. The options to run the service displayed by the 'Interface Program'

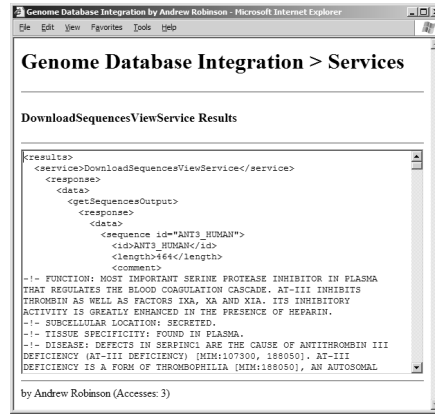


Fig. 14. Results of a service displayed by the 'Interface Program'

5.2 The Mediator

One of the core parts to the Mediator is the conversion between the external proprietary data structures to the internally used 'revised BSML'. The second phase of mapping process has been implemented as a Document Object Model (DOM) which was based on the structure of the 'Data Elements' and thus named 'Data Element-DOM' (DE-DOM). The first phase of the mapping process parses the data from the input format and loads it into the DE-DOM. Once the entire sequence is loaded the 'produceXML' method is called on the DE-DOM to produce the output BSML. As you might note the conversion is done one sequence at a time, this allows the real time display of returned sequences to the user and could allow the user to abort the process if the results don't look right.

The mediator structure has also been implemented in the object-oriented manner. Each layer and its modules are implemented as a single java class as shown in fig 3.

6 Conclusion

In the earlier chapters we have presented a solution to the genome database integration problem. The main needs for this are (1) an architecture that provides the easy addition of databases and formatting / manipulation modules (2) A common 'language' that supports the interchange of data and (3) a platform that uses existing standard or common practices in data format. The use of a mediation layer which has three internal layers addresses the first point noted above. Points 2 and 3 above have been addressed by using BSML data format since it can be used as the common language and it is also based on the XML standard.

Future work will be to optimise the efficiency of the mapping process and to develop a 'smart' user interface so that is more useable by the target users.

References

1. Andrew J. Robinson (2003) Honours thesis: Genome database integration using XML and mediators, Dept. Computer Science and Computer Engineering LaTrobe University.
2. T. K. Attwood, M. D. R. Croning, D. R. Flower, A. P. Lewis, J. E. Mabey, P. Scordis, J. N. Selly and W. Wright (2000) PRINTS-S the database formerly known as PRINTS, *Nucleic Acids Research*, pp. 225–227.
3. Yury V. Bukhman and Jeffrey Skolnick (2001) BioMolQuest: integrated database-based retrieval of protein structural and functional information, *Bioinformatics* Vol. 12 no. 5, pp. 468–478.
4. Norman W. Paton, Shakeel A. Hhan, Andrew Hayes, Fouzia Moussouni, Andy Brass, Karen Eilbeck, Carole A. Goble, Simon j. Hubbard and Stephen G. Oliver (2000) Conceptual modelling of genomic information, *Bioinformatics*, pp. 548–557.
5. Jinyan Li, See-Kiong Ng, and Limsoon Wong (2003) Bioinformatics Adventure in Database Research, *LNCS 2472,* pp. 31–46.
6. Luiz Fernando Bessa Seibel and Sergio Lifschitz (2001) A Genome Databases Framework, *LNCS 2113,* pp. 319–329.

7. Robert Stevens, Patricia Baker, Sean Bechhofer, Gary Ng, Alex Jacoby, Norman W. Paton, Carole A. Goble and Andy Brass (2000) TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources, *Bioinformatics,* pp. 184–185.
8. P. Buneman, S. B. Davidson, K. Hart, C. Overton and L. Wong (1995) A Data Transformation System for Biological Data Sources. *Proceedings of the 21st VLDB Conference,* pp. 158–169.
9. Stephen Chan, Tharam Dillon, Andrew Siu (2002) Applying a mediator architecture employing XML to retailing inventory control, *The Journal of Systems and Software 60,* pp. 239–248.
10. http://www.ebi.ac.uk/xembl/, *The XEMBL Project: the EMBL/GenBank/DDBJ databases in XML*, European Bioinformatics Institute.
11. http://www.w3.org/XML/, *Extensible Markup Language (XML)*, World Wide Web consortium.

Protein Structure Prediction with Stochastic Optimization Methods: Folding and Misfolding the Villin Headpiece

Thomas Herges, Alexander Schug, and Wolfgang Wenzel

Forschungszentrum Karlsruhe, Institut für Nanotechnologie,
Postfach 3640, D-76021 Karlsruhe, Germany

Abstract. We recently developed an all-atom free energy forcefield (PFF01) for protein structure prediction with stochastic optimization methods. Using this forcefield we were able to reproducibly fold the 20 amino-acid trp-cage protein and the 40-amino acid three-helix HIV accessory protein. We could also demonstrate that PFF01 stabilized the native folds of various other proteins ranging from 40-60 amino acids at the all atom level. Here we report on a folding study on the widely investigated autonomously folding 36-amino acid villin headpiece. Using more than 76000 low-energy decoys to characterize its free-energy landscape, we find several competing low-lying three-helix structures. The existence of these metastable conformations, which are not nearly as prevalent in other proteins, may explain the extreme difficulty in folding this protein in-silico.

1 Introduction

Ab-initio protein tertiary structure prediction (PSP) and the elucidation of the mechanism of the folding process are among the most important outstanding problems of biophysical chemistry [1,2,3]. The many complementary proposals for PSP span a wide range of representations [4,5,6,7,8,9,10,11,12] of the protein conformation, ranging from coarse grained models to atomic resolution. The choice of representation often correlates with the methodology employed in structure prediction, ranging from empirical potentials for coarse grained models [4, 6,12] to complex atom-based potentials that directly approximate the physical interactions in the system [13,14]. The latter offer insights into the mechanism of protein structure formation and promise better transferability [15], but their use incurs large computational costs that has confined all-atom protein structure prediction to all but the smallest peptides [14,16]. Recent evaluations of different approaches to PSP found that empirical, homology based methods outperform *ab-initio* methods [17,18], but that there is still much room for improvement in the quality of the predictions of all methods.

It has been one of the central paradigms of protein folding that proteins in their native conformation are in thermodynamic equilibrium with their environment [19]. Exploiting this characteristic the structure of the protein

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 454–464, 2004.
© Springer-Verlag Berlin Heidelberg 2004

can be predicted by locating the global minimum of its free energy surface [11, 20] without recourse to the folding dynamics, a process which is potentially much more efficient than the direct simulation of the folding process. Direct simulation by molecular dynamics directly elucidates the folding dynamics of the protein but strains presently available computational resources and will remain inapplicable to large proteins in the foreseeable future. For many applications that require only the folded structure of the protein, PSP based on global optimization of a the free energy may offer a viable alternative approach, provided that suitable parameterization of the free energy of the protein in its environment exists and that global optimum of this free energy surface can be found with sufficient accuracy [21].

PSP by optimization requires an accurate, yet tractable model for the low-energy portion of the free energy surface of the protein and efficient optimization techniques which can reliably determine the its global minimum. Despite a steady increase in available computational power PSP remains one of the grand computational challenges, which constrains the choice of free energy parameterizations. The free energy model must describe the internal energy of the system the entropy of the both the molecule and the surrounding solvent. It has long been argued that entropy differences between near-native conformations of the protein are dominated by changes in the free energy of the solvent (and possibly the sidechains). Changes in the solvent free energy may be approximated by simple models based on the solvent accessible surface area (SASA) of the individual atoms [22] in an implicit solvent model. Such terms also approximately account for the changes in the sidechain entropy upon burial and other non-bonded interactions with fictitious solvent molecules.

We have recently demonstrated a feasible strategy for all-atom protein structure prediction [23,24,25] in a minimal thermodynamic approach. We developed an all-atom free-energy forcefield for proteins (PFF01), which is primarily based on physical interactions with important empirical, though sequence independent, corrections [25]. We already demonstrated the reproducible and predictive folding of two proteins, the 40 amino acid HIV accessory protein (1F4I) [24] and the 20 amino acid trp-cage protein (1L2Y) [23] with PFF01. In addition we could show that PFF01 stabilizes the native conformations of three other proteins: the 52 amino-acid protein A [14,26], which folds into a three helix bundle [7, 27,28]; the engrailed homeodomain (1ENH) from *Drosophilia melangster* [29], a prototypical 54-residue α helical protein [30] and the bacterial ribosomal protein L20 (1GYZ), an 60-amino acid four-helix protein [31].

In this study we investigate the free-energy landscape of the autonomously folding 36 amino acid headpiece of the villin protein (pdb-code 1VII), which did not fold in all-atom simulations using the AMBER [13] and ECEPP/2 [32] forcefields. We resolved the entire low-energy free-energy surface (FES) of the protein using a decoy-tree approach. This analysis demonstrante 1VII has several metastable three-helix conformations, which have almost the same energy and secondary structure content as the native state, but which differ significantly in tertiary structure from the NMR conformation. We then attempted to fold the villin headpiece from random initial conformations. While

over 50% of the runs found conformations associated with the folding funnel leading towards the native state, a near native conformation emerged only as second best in the folding run. The best conformation found in the folding run, however, had a higher energy than the best NMR decoy of the free energy surface. These results illustrate that the bottleneck in further progress for all-atom protein folding lies in the availability of efficient optimization methods. Although the villin headpiece is smaller than the HIV accessory protein, which was sucessfully folded reproducibly with the same methodology, its free energy surface is significantly more complex.

2 Methods

2.1 Model

We have recently developed an all-atom (with the exception of apolar CH_n groups) free-energy protein forcefield (PFF01) that models the low-energy conformations of proteins with minimal computational demand [33,25]. In the folding process at physiological conditions the degrees of freedom of a peptide are confined to rotations about single bonds. The forcefield is parameterized with the following non-bonded interactions:

$$V(\{\boldsymbol{r}_i\}) = \sum_{ij} V_{ij} \left[\left(\frac{R_{ij}}{r_{ij}} \right)^{12} - \left(\frac{2R_{ij}}{r_{ij}} \right)^{6} \right] + \sum_{ij} \frac{q_i q_j}{\epsilon_{g(i)g(j)} r_{ij}} + \sum_i \sigma_i A_i + \sum_{\text{hbonds}} V_{hb}. \tag{1}$$

Here r_{ij} denotes the distance between atoms i and j and g(i) the type of the amino acid i. The Lennard Jones parameters (V_{ij}, R_{ij} for potential depths and equilibrium distance) depend on the type of the atom pair and were adjusted to satisfy constraints derived from as a set of 138 proteins of the PDB database [34, 33,35]. The non-trivial electrostatic interactions in proteins are represented via group-specific dielectric constants (ϵ_i depending on the amino-acid to which atom i belongs). The partial charges q_i and ϵ_i were previously derived in a potential-of-mean-force approach [36]. Interactions with the solvent were first fit in a minimal solvent accessible surface model [22] parameterized by free energies per unit area σ_i to reproduce the enthalpies of solvation of the Gly-X-Gly family of peptides [37]. A_i corresponds to the area of atom i that is in contact with a ficticious solvent. The σ_i were adjusted to stabilize the native state of the 36-amino acid headgroup of villin (pdb-code 1VII) as the global minimum of the forcefield [38]. Hydrogen bonds are described via dipole-dipole interactions included in the electrostatic terms and an additional short range term for backbone-backbone hydrogen bonding (CO to NH) which takes the form: $V_{hb}(CO_i, NH_j) = R(\tilde{r}_{ij})\Gamma(\phi_{ij}, \theta_{ij})$ where $\tilde{r}_{ij}$,ϕ_{ij} and θ_{ij} designate the OH distance, ϕ is the angle between N,H and O along the bond and θ is the angle between the CO and NH axis. R and Γ were fitted as a corrective potentials of mean force to the same set of proteins described above [34,25].

2.2 Optimization Method

Monte-Carlo with minimization (MCM) has been used to locate the global minima of many complex potential energy surfaces [21,39,40,24,25]. The minimization step simplifies the potential energy surface (PES), by mapping each conformation to a nearby local minium. The increase of efficiency of MCM in comparison to the Monte-Carlo method (MC) [41] on the original potential energy surface strongly depends on the average energy gain in the minimization procedure. For very rugged potential energy surfaces, such as those encountered in protein folding, local minimization yields comparatively little improvement. We have therefore replaced the local minimization by a simulated annealing (SA) [42] run, starting at 660 K and then cooled with a geometric cooling cycle to 1 K. The number of steps in the cooling cycle is gradually increased according to $N_c = 10^5\sqrt{n_m}$ with the number of the minimization cycle n_m. The resulting configuration replaces the starting configuration according to a threshold acceptance criterion with a threshold of 3 kcal/mol. During the SA simulations, changes in the dihedral angles are generated randomly for both sidechain and main-chain dihedral angles. Global moves for the main-chain dihedral angles are additionally generated from a library [43].

2.3 Free Energy Surface Topology

The topology of the low-energy part of the FEs was analyzed in a decoy tree [44, 45] that groups conformations in a given energy range into families, as a distance measure we used backbone RMS deviation (RMSB). The tree was constructed from all decoys (local minima that differ by at least 1 Å RMSB from all other decoys) encountered in the simulations for sequence of equidistant energies $E_0, E_1 \ldots$, starting with the energy of the best conformation. A decoy with energy below E_n that has less than 3 Å RMSB to the decoy of just one family at the next lower energy level E_{n-1} is included into that family. If a decoy is associated with more than one family, the corresponding families are united, if it belongs to no existing family a new family containing just this decoy is created. For each family we by draw a vertical line in the energy window between E_{n-1} and E_n and merge the lines for the energy E_n where the the families are united. This analysis results in an inverted tree-like structure that illustrates the energetic order and degree of structural similarity of conformations via their family association. For a forcefield that stabilizes the native structure the native family is represented by the branch of the tree that extends the furthest downward. In forcefield that stabilize non-native structures, perturbations in the parameters rebalance the lower portion of the tree.

3 Results

In order to analyze the low-energy portion of the free-energy landscape of the villin headpiece we generated in excess of 76000 structures with the modified

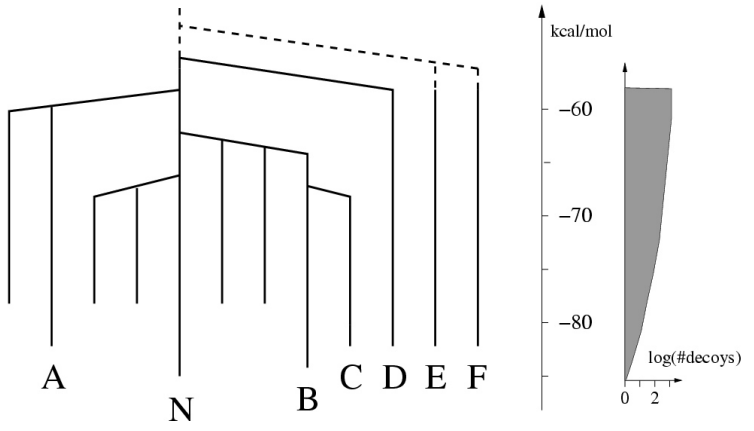


Fig. 1. Decoy tree for the villin headpiece in the optimized forcefield. The structures corresponding to the terminal branches of the tree are shown in Figure (2). The tree was constructed from a set of 76000 structures grouped in 14000 families as discussed in methods.

basis hopping method described above. These were grouped in to 14,000 decoy families, according to the procedure outline in the methods section. The resulting free energy surface is illustrated by the decoy tree shown in Figure (1). The variation of the solvation parameters lead to a flattening of the FES, which now has seven almost isoenergetic non-native terminal branches. The decoy associated with the terminal branches of the tree are shown in Figure (2). There are now five three helix and two two-helix structures at the bottom of their respective branches of the tree. One notes a distinct propensity for the formation of helices in those regions where the NMR structure forms helices 1 and 3. The central helix occurs less often and leads to misfolded decoys reminiscent to the two-helix structures found in [46]. The energy spectrum shows a gap of only 1 kcal/mol between the NMR decoy and the next competing structure and becomes nearly continuous with increasing energy. An analysis of the RMSB matrix of all low energy decoys suggests that many distinct metastable conformations were probed in the search.

Having characterized the low-energy free-energy surface of the protein we attempted to fold 1VII from random initial structures following the strategy applied in the folding to the HIV accessory protein [24]. We first performed 500 independent Monte-Carlo simulations on random unfolded conformations of the villin headpiece, each comprising 2×10^5 energy evaluations. The resulting structures were ranked according to their energy. We then selected a set of starting configurations for the folding simulations beginning with the best energy. We then selected 49 further conformations with increasing energy, provided that the selected conformation differed by at least 4Å in the backbone root mean square deviation (RMSB) from all conformations selected so far.

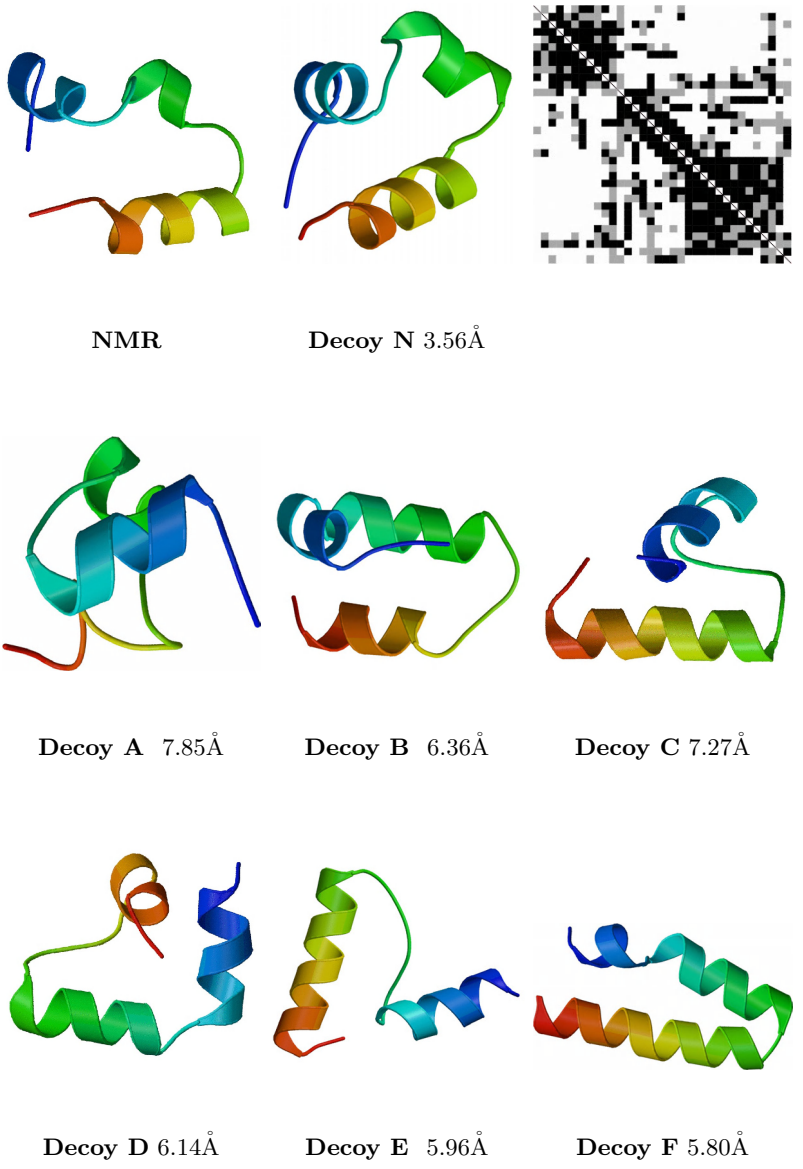


Fig. 2. Native structure (labeled NMR) and the conformations of the terminal branches of the decoy tree of the villin headpiece. The labels refer to the the branches in Figure (1) and the numbers indicate the RMSB deviations to the native structure. The color coded distance map in the top right compares the C_β-C_β distances of the native structure with that of the NMR decoy. A pixel in row i and colum j of the figure indicates the difference in the C_β-C_β distances of the native and the compared structure. Black (grey) squares indicate that the C_β-C_β distances of the native and the other structure differ by less than 1.5 (2.25) Å respectively. White squares indicate larger deviations.

Table 1. Top 20 decoys after the initial simulations of the villin headpiece in PFF01 (see text) with their energy (in kcal/mol), the RMSB deviation to the native structure, the closest terminal branch of the tree (see text) and the RMSB deviation between the terminal branch and the decoy. Next we show the label of the conformation of the closest branch in the decoy tree Figure (1) and the RMSB deviation between the two conformations. An 'X' donates a conformation that hat larger than 5 Å RMSB deviation from all branches of the tree. The last column shows the secondary structure content (computed with DSSP).

Code	Energy	RMSB	Branch	Dist	Secondary Structure
NMR	—				CHHHHHTTSSSCHHHHTTSCHHHHHHHHHHTTCC
D01	-86.23	7.57	C	(3.31)	CHHHHHHHHHHCHHHHHHSHHHHHHHHHHHTCC
D02	-85.51	4.56	N	(3.27)	CHHHHHHHTSCHHHHHHCHHHHHHHHHHHTCC
D03	-85.35	5.80	N	(4.86)	CSHHHHHHHHHCHHHHHHCHHHHHHHHHTTTCCC
D04	-85.11	4.30	N	(3.23)	CHHHHHHHTSCSHHHHHCHHHHHHHHHHHTCC
D05	-84.08	4.13	N	(3.35)	CHHHHHHHTSCHHHHHHSHHHHHHHHHHHTCC
D06	-83.46	7.70	C	(3.14)	CHHHHHHHHHHCSSCSSCHHHHHHHHHHHTCC
D07	-82.97	4.62	N	(3.37)	CHHHHHHHCHHHHHHHHSHHHHHHHHHHHHSCC
D08	-82.55	5.93	N	(4.65)	CHHHHHHHHHHHCHHHHHSCTTTCHHHHHHHHHC
D09	-82.03	6.50	A	(4.42)	CCCHHHHHHHTCSCHHHHHHSSSHHHHHHHHHHT
D10	-81.07	6.76	E	(4.08)	CCCHHHHHHHTSSCCSSCSSHHHHHHHHHHHHHT
D11	-80.96	7.42	A	(4.00)	CHHHHHHHTCSCHHHHHHSCSHHHHHHHHHHTCC
D12	-80.38	7.11	X		CHHHHHHHHHHCCSHHHHCSSHHHHHHHHHHTCC
D13	-80.36	7.56	C	(3.49)	CHHHHHHHHHHHTTCCSSSHHHHHHHHHHHHTCC
D14	-79.94	8.29	X		CHHHHHHHHHTTTSSCSCSSHHHHHCHHHHHHHT
D15	-79.85	3.86	N	(4.54)	CHHHHHHHHHTCSCHHHHHHSCHHHHHHHHHHTS
D16	-79.25	5.03	N	(3.48)	CHHHHHHHHHCSSCHHHHHSHHHHHHHHHHHHHT
D17	-78.74	2.66	N	(2.75)	CHHHHHHHTSCCHHHHHHSCHHHHHHHHHHHTCC
D18	-78.39	7.58	C	(3.59)	CHHHHHHHHHCCCHHHHHSHHHHHHHHHHHTCC
D19	-78.35	3.18	N	(3.49)	CHHHHHHHTSCCHHHHHHSCHHHHHHHHHHHHHC
D20	-78.21	7.30	B	(4.00)	CHHHHHHHHHHHHHHHHHSHHHHHHHHHHHTCC

Starting from these conformations, we performed 50 independent basin hopping runs, as described in the methods section, for 100 cycles each, resulting in 67×10^6 energy evaluations for each decoy. The 20 resulting structures with the best energy are summarized in Table (1), along with their RMSB deviation to the NMR conformation and their secondary structure content. We note that the low-energy decoys are very close to one another and that – unfortunately – a non-native conformation emerges as the lowest decoy. We also note that the the best conformation (decoy D01) fails to reach the energy of the NMR decoy of the decoy tree (-86.55) kcal/mol) in Figure (1). The failure to fold is thus attributable to a deficiency in the optimization method, rather than to deficiencies of the forcefield. Apparently the threshold acceptance criterion in the adapted basin hopping method leads to energy fluctuations larger than the energy differences between the decoys.

In comparison to earlier studies [47,46], it is encouraging that three-helix structures with the right secondary structure dominate the low-energy decoys.

The RMSB values in the table also clearly demonstrates that obtaining the correct secondary structure is a necessary, but not a sufficient condition for proper folding of the protein. In order to further analyze the results of the folding runs we have computed the RMSB distance matrix between the conformations associated with the terminal branches of the decoy tree (Figure (1)), as depicted in Figure (2) and the conformations D01-D20. Table (1) shows the label of the closest branch for each decoy and its RMSB deviation to the bottom structure. Out of the 20 simulations the native branch (N) was visited 11 times, branch C was visited four times, branch A twice and branches C and B once. One conformation (labeled X) could not be associated with any terminal branch conformation, which is not surprising because high energy conformations should be compared to conformations of corresponding energy. This identification of the folding simulations helps us to analyze the optimization method: it is encouraging that the adapted basin hopping method is able to locate the folding funnel leading to the native conformation in over 50% of all simulations. Its energy resolution, however, appears to be insufficient to locate the local minimum associated with this funnel despite the enormous CPU resources invested. In order to check the convergence we performed additional simulations of 40 basin hopping cycles, comprising 50×10^6 energy evaluations (17 CPU days on a 2.4 Ghz Intel XEON processor), of the lowest seven decoy structures. These demonstrated that up to 2 kcal/mol can be gained in the total energy of each structure, without significantly changing the energetic order of the conformations, their branch association or their RMSB deviation to the NMR structure. As a result, one may assume that the overall effort to predictively fold 1VII with the basin hopping technique may be even larger.

4 Discussion

Recent progress in all-atom protein structure prediction with free-energy models [23,28,24,25] provides increasing evidence for the thermodynamic hypothesis [19] that protein tertiary structure can be predicted at the all atom level as the equilibrium conformation of suitable free-energy forcefield. We find that PFF01 stabilizes the native conformation of a family of non-homologous helical proteins as its global minimum. Our all-atom representation of the protein permits the parameterization of the free energy landscape on the basis of physical interactions that are well understood for smaller systems. Parameterizations based on physical interactions promise transferability, but our experience suggests that some heuristic corrections for complex interactions, e.g. screening effects of the implicit solvent and hydrogen bonding may be required.

The use of physical interactions in all-atom representations incurs a large computational cost, when compared to more coarse grained, or homology based models. In an optimization approach this increase in cost is partially compensated by the efficiency of the conformational search. In comparison to folding by molecular dynamics, the disadvantage of an optimization strategy is the loss of dynamical information. With respect to protein structure prediction, a major

advantage of optimization methods of the free-energy lies in their predictivity which can presently not be claimed for all but the smallest peptides [47] with direct simulation techniques. Even so, the energy resolution of the optimization method, rather than inaccuracies of the forcefield, appear to be the bottleneck towards progress for larger proteins.

The villin headpiece that we attempted to investigate here provides an interesting example for this approach: while we were able to fully characterize the low energy free-energy surface of the protein, reproducible folding, which could be achieved for the trp-cage protein [23] and the HIV accessory protein [24], presently overtaxes the energy resolution of the basin hopping method. The close proximity of the different branches of decoy tree may provide a rationalization of the difficulties encountered in prior folding studies, which also failed to converge to the native structure [47,46].

One cannot overemphasize the importance of the interplay of optimization methods and forcefield validation. Rational forcefield development mandates the ability to generate decoys that fully explore competing low-energy conformations to the native state. The success of different optimization strategies depends strongly on the structure of the potential energy surface. As a result the development of efficient optimization techniques for all-atom protein structure prediction depends on the availability of a forcefield that folds proteins with appreciable hydrophobic cores. For helical proteins the bottleneck in *ab-initio* all-atom structure prediction now lies in the development of optimization strategies that significantly increase the system size that can be treated with present day computational resources. We note that PSP on the basis of forcefield optimization fits the computational paradigm of globally distributed grid computing even better than protein folding using a molecular dynamics approach.

We hope that the rational strategy of all-atom forcefield optimization pursued in this projects offers a valuable contribution to protein structure prediction in general. Even if the available computational resources remain insufficient in the foreseeable future to fold large proteins, the all atom forcefields may be used to discriminate among conformations generated on the basis of other techniques, in particular with coarse grained, or knowledge based potentials [12].

Acknowledgements. We thank the Fond der Chemischen Industrie, the BMBF, the Deutsche Forschungsgemeinschaft (grants WE 1863/10-2,WE 1863/14-1) and the Kurt Eberhard Bode Stiftung for financial support.

References

1. Baker, D., Sali, A.: Protein structure prediction and structural genomics. Science **294** (2001) 93
2. Moult, J., Fidelis, K., Zemia, A., Hubbard, T.: Critical assessment of methods of protein structure (casp): round iv. Proteins **45** (2001) 2–7
3. Schonbrunn, J., Wedemeyer, W.J., Baker, D.: Protein structure prediction in 2002. Curr. Op. Struc. Biol. **12** (2002) 348–352

4. Go, N., Scheraga, H.A.: On the use of classical statistical mechanics in the treatment of polymer chain conformation. Macromolecules **9** (1976) 535–542
5. Nemethy, G., Gibson, K.D., Palmer, K.A., Yoon, C.N., Paternali, G., Zagari, A., Rumsey, S., Scheraga, H.: Energy parameters in polypeptides 10. improved geometrical paramters and nonbonded interactions for use in the ecepp/3 algorithm. J. Phys. Chem. **96** (1992) 6472–6484
6. Ulrich, P., Scott, W., W. F. van Gunsteren, W., Torda, A.E.: Protein structure prediction forcefields: Paramterization with quasi newtonian dynamics. Proteins, SF&G **27** (1997) 367–384
7. Zhou, Y., Karplus, M.: Folding a model three helix bundle protein: thermodynamic and kinetic analysis. J. Molec. Biol. **293** (1999) 917
8. Simons, K.T., Kooperberg, C., Huang, E., Baker, D. J. Molec. Biol. **286** (1997) 209–225
9. Simons, K.T., Ruczinski, I., Kooperberg, C., Fox, B., Bystroff, C., Baker, D. Proteins, SF&G **34** (1999) 82–95
10. Pillardy, J., Czaplewski, C., Liwo, A., andD. R. Ripoll, J.L., Kamierkiewicz, R., andW. J. Wedemeyer, S.O., Gibson, K.D., Arnautova, Y.A., Saunders, J., Ye, Y.J., Scheraga, H.A.: Recent improvements in prediction of protein structure by global optimization of a potential energy function. Proc. Natl. Acad. Sci.(USA) **98** (2001) 2329
11. Liwo, A., Arlukowicz, P., Czaplewski, C., Oldizeij, S., Pillardy, J., Scheraga, H.: A method for optimising potential energy functions by a hierarchichal design of the potential energy landscape. Proc. Natl. Acad. Sci.(USA) **99** (2002) 1937–1942
12. Nanias, M., Chinchio, M., Pillardy, J., Ripoll, D.R., Scheraga, H.A.: Packing helices in proteins by global optimization of a potential energy function. Proc. Natl. Acad. Sci.(USA) **100** (2003) 1706–1710
13. Duan, Y., Kollman, P.A.: Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. Science **282** (1998) 740
14. Snow, C.D., Nguyen, H., Panda, V.S., Gruebele, M.: Absolute comparison of simulated and experimental protein folding dynamics. Nature **420** (2002) 102–106
15. Vasquez, M., Nemethy, G., Scheraga, H. Chem. Rev. **94** (1994) 2138–2239
16. Simmerling, C., Strockbine, B., Roitberg, A.: All-atom strucutre prediction and folding simulations of a stable protein. J. Am. Chem. Soc. **124** (2002) 11258
17. Lattman, E.: Casp4. Proteins **44** (2001) 399
18. Bonneau, R., Tsui, J., Ruczinski, I., Chivian, D., Strauss, C.M.E., Baker, D.: Rosetta in casp4: progress in ab-initio protein structure prediction. Proteins **45** (2001) 119–126
19. Anfinsen, C.B. Science **181** (1973) 223–230
20. Onuchic, J.N., Luthey-Schulten, Z., Wolynes, P.G.: Theory of protein folding: The energy landscape perspective. Annu. Rev. Phys. Chem. **48** (1997) 545–600
21. Li, Z., Scheraga, H.: Monte carlo minimization approach to the multiple minima problem in protein folding. Proc. Nat. Acad. Sci. U.S.A. **84** (1987) 6611
22. Eisenberg, D., McLachlan, A.D.: Solvation energy in protein folding and binding. Nature **319** (1986) 199–203
23. Schug, A., Herges, T., Wenzel, W.: Reproducible protein folding with the stochastisc tunneling method. Phys. Rev. Letters **91** (2003) 158102
24. Herges, T., Wenzel, W.: Reproducible in-silico folding of a three-helix protein in a transferable all-atom forcefield. http://www.arXiv.org: physics/0310146 (2004) (submitted to Phys. Rev. Lett.).
25. Herges, T., Wenzel, W.: Development of an all-atom forcefield for teriatry structure prediction of helical proteins. (submitted to Proteins) (2004)

26. Gouda, H., Torigoe, H., Saito, A., Sato, M., Arata, Y., Shimanda, I. Biochemistry **31** (1992) 9665–9672
27. Schea, J., Onuchic, J., Brooks III, C. Proc. Natl. Acad. Sci.(USA) **96** (1999) 12512–12517
28. Vila, J., Ripoll, D., Scheraga, H.: Atomically detailed folding simulation of the b domain of staphylococcal protein a from random structures. Proc. Natl. Acad. Sci.(USA) **100** (2004) 14812–14816
29. Clarke, N., Kissinger, C., J.Desjarlais, Gilliland, G., Pabo, C.: Structural studies of the engrailed homedomain. Prot. Sci. **3** (1994) 1779–1787
30. Mayor, U., Guydosh, N.R., Johnson, C.M., Grossmann, J.G., Sato, S., Jas, G.S., Freund, S.M.V., Alonso, D.O.V., Daggett, V., Fersht, A.R.: The complete folding pathway of a protein from nanoseconds to micorseconds. Nature **421** (2003) 863–867
31. Raibaud, S., Lebars, I., Guillier, M., Chiaruttini, C., Bontems, F., Rak, A., Garber, M., Allemand, F., Springer, M., Dardel, F.: Nmr structure of bacterial ribosomal protein l20: Implications for ribosome assembly and translational control. JMB **323** (2002) 143
32. Hansmann, H.: Global optimization by energy landscpae paving. Phys. Rev. Letters **88** (2002) 068105
33. Herges, T., Merlitz, H., Wenzel, W.: Stochastic optimisation methods for biomolecular structure prediction. J. Ass. Lab. Autom. **7** (2002) 98–104
34. Abagyan, R., Totrov, M.: Biased probability monte carlo conformation searches and electrostatic calculations for peptides and proteins. J. Molec. Biol. **235** (1994) 983–1002
35. Herges, T., Schug, A., Burghardt, B., Wenzel, W.: Low energy conformations of a three helix peptide in an all-atom biomolecular forcefield. Intl. J. Quantum Chem. (2003) (in press)
36. Avbelj, F., Moult, J.: Role of eloctrostatic screening in determining protein main chain conformational preferences. Biochemistry **34** (1995) 755–764
37. Sharp, K.A., Nicholls, A., Friedman, R., Honig, B.: Extracting hydrophobic free energies from experimental data:relationship to protein folding and theoretical models. Biochemistry **30** (1991) 9686–9697
38. Herges, T., Schug, A., Merlitz, H., Wenzel, W.: Stochastic optimization methods for structure prediction of biomolecular nanoscale systems. Nanotechnology **14** (2003) 1161–1167
39. Doyle, J.P., Wales, D. J.Chem.Phys. **105** (1996) 8428
40. Wales, D.J., Doyle, J.P. JPC **101** (1997) 5111
41. Metropolis, N., andM. N. Rosenbluth, A.W.R., Teller, A.H., Teller, E.: Equation of state calculations by fast computingmachines. J. Chem. Phys. **21** (1953) 1087–1092
42. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **220** (1983) 671–680
43. Avbelj, F., Moult, J. Proteins **23** (1995) 129–141
44. Brooks, C.L., Onuchic, J.N., Wales, D.J.: Taking a walk on a landscape. Science **293** (2001) 612
45. Becker, O., Karplus, M. JCP **106** (1997) 1495
46. Lin, C., Hu, C., Hansmann, U.: Parallel tempering simulations of hp-36. Proteins **53** (2003) 436–445
47. Daura, X., Juan, B., Seebach, D., van Gunsteren, W., Daura, A.E.M., Juan, B., Seebach, D., van Gunsteren, W., Mark, A.E. J. Molec. Biol. (1998) 925

High Throughput *in-silico* Screening against Flexible Protein Receptors

Holger Merlitz and Wolfgang Wenzel

Forschungszentrum Karlsruhe,
Institut für Nanotechnologie,
Postfach 3640, D-76021 Karlsruhe, Germany
email:wenzel@int.fzk.de
http:www.fzk.de/biostruct

Abstract. We report results for the in-silico screening of a database of 10000 flexible compounds against various crystal structures of the thymidine kinase receptor complexed with 10 known substrates. The ligands were docked using *FlexScreen*, a recently developed docking tool based on the stochastic tunneling method. We used a first-principle based scoring function. For rigid receptor conformations we find large deviations in the rank of the known inhibitors depending on the choice of receptor conformation. These data demonstrate that the failure to dock originates from the neglect of receptor degrees of freedom and is not attributable to deficiencies in the scoring function or the docking algorithm. We then performed a screen in which critical receptor sidechains were permitted to change their conformation and found improved scores for those inhibitors that did not dock well in any of the previous screens. Consequently, the consideration of receptor sidechain flexibility in all-atom *FlexScreen* improves the quality of the screening approach.

1 Introduction

Virtual screening of chemical databases to targets of known three-dimensional structure is developing into an increasingly reliable method for finding new lead candidates in drug development [1,2]. Both better scoring functions [3] and novel docking strategies [4] contribute to this trend, although no completely satisfying approach has been established yet [5]. This is not surprising since the approximations which are needed to achieve a reasonable screening rates impose significant restrictions on the virtual representation of the physical system. Relaxation of these restrictions, such as permitting ligand or receptor flexibility, potentially increase the reliability of the scoring process, but come at a high computational cost.

The limitations of presently available computational resources and the large number of possible ligands enforce severe approximations in the representation of receptor and ligand, and their interactions. Significant computational efficiency is gained, when the protein receptor is assumed to be rigid in the docking process; for this reason many tests of screening functions [5] and virtually all

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 465–472, 2004.
© Springer-Verlag Berlin Heidelberg 2004

large scale computational screens presently rely on a rigid-receptor conformation. On the other hand, direct comparison between ligand-free and complexed crystal structures often demonstrate a significant ligand-induced alteration of the receptor structure.

While ligand flexibility is now routinely considered in many atomistic in-silico screening methods, accounting for receptor flexibility still poses significant challenges [6,7,8]. The thymidine kinase receptor is a useful benchmark system for the evaluation of screening methods, because not just one, but several substrates are known and characterized in the their binding mode. Here we use this system as a prototypical example to document the shortcoming of rigid receptor screens, independent of the particular choice of the receptor conformations. We then present screens using *FlexScreen*[9,10], a recently developed all-atom screening tool based on the stochastic tunneling method to screen a subset of up to 10000 ligands of the NCI-Open database considering receptor sidechain flexibility.

In the following we first describe the two main ingredients to *all-atom in-silico screening*: the docking tool *FlexScreen* and the parameterization of the scoring function that approximates the binding energy of the ligand to the receptor. Next we present the results of several screens of 10000 ligands of the NCI database against specific rigid receptor conformations and introduce a scoring scheme that quantifies the quality of a particular screen. Finally we perform a screen with a *flexible receptor* and discuss its advantages compared to the previous rigid receptor screens.

2 Methods

There are two major ingredients to an all-atom in-silico screening method: (1) a scoring function that approximates the binding energy (ideally the affinity) of the receptor-ligand complex as a function of the conformation of this complex. and (2) an efficient optimization method that is able to locate the binding mode of a given ligand to the receptor as the global optimum of the scoring function. In a database screen, all ligands are thus assigned an optimal score which is then used to sort the database to select suitable ligands for further investigations.

The screens in this investigation were performed with *FlexScreen*, an all-atom docking approach[9,10] based on the stochastic tunneling method [11]. This method was shown to be superior to other competing stochastic optimization methods[9] and had performed adequately in a screening of 10000 ligands to the active site of dihydrofolate reductase (pdb code 4dfr [12]), where the known inhibitor (methotrexate) emerged as the top scoring ligand [10].

The stochastic tunneling technique was proposed as a generic global optimization method for complex rugged potential energy surfaces(PES). In STUN the dynamical process explores not the original, but a transformed PES, which dynamically adapts and simplifies during the simulation. For the simulations reported here we replace the original transformation [11] with:

$$E_{STUN} = \ln\left(x + \sqrt{x^2 + 1}\right) \tag{1}$$

where $x = \gamma(E - E_0)$. E is the energy of the present conformation and E_0 the best energy found so far. The problem-dependent transformation parameter γ controls the steepness of the transformation [11]. The general idea of this approach is to flatten the potential energy surface in all regions that lie significantly above the best estimate for the minimal energy (E_0). Even at low temperatures the dynamics of the system becomes diffusive at energies $E \gg E_0$ independent of the relative energy differences of the high-energy conformations involved. The dynamics of the conformation on the untransformed PES then appears to "tunnel" through energy barriers of arbitrary height, while low metastable conformations are still well resolved. Applied to receptor-ligand docking this mechanism ensures that the ligand can reorient through sterically forbidden regions in the receptor pocket.

Many different scoring functions have been proposed in recent years [3] and no clear consensus has emerged to date on the superiority of physics-based or knowledge based approaches. In this investigation we employed the a simple, first-principle scoring function:

$$S = \sum_{Protein} \sum_{Ligand} \left(\frac{R_{ij}}{r_{ij}^{12}} - \frac{A_{ij}}{r_{ij}^{6}} + \frac{q_i q_j}{r_{ij}} \right) + \sum_{h-bonds} \cos \Theta_{ij} \left(\frac{\tilde{R}_{ij}}{r_{ij}^{12}} - \frac{\tilde{A}_{ij}}{r_{ij}^{10}} \right), \quad (2)$$

which proved successful in a prior investigation of the dihydrofolate reductase receptor [10]. It contains the empirical Pauli repulsion (first term), the Van de Waals attraction (second term), the electrostatic potential (third term) and an angular dependent hydrogen bond potential (term four and five). The Lennard-Jones parameters R_{ij} and A_{ij} were taken from OPLSAA [13], the partial charges q_i were computed with InsightII and esff force field, and the hydrogen bond parameters $\tilde{R}_{ij}$, $\tilde{A}_{ij}$ were taken from AutoDock [14]. This force field lacks solvation terms to model entropic or hydrophobic contributions. The omission of such terms has been argued to be appropriate for constricted receptor pockets in which all ligands with high affinity displace essentially all water molecules. Each screen was repeated 6 times to reduce the influence of statistical fluctuations and the best affinity was used for ranking the ligands.

3 Results

3.1 Rigid Receptor Screens

We investigated the degree of database enrichment of 10000 compounds, randomly chosen from the nciopen3D database [15], and 10 known substrates when docked to the X-ray TK receptor structure, which was experimentally determined in complex with one of the substrates, dt (deoxythymidine, pdb entry 1ki2 [16]). In this screen 5353 ligands attained a stable conformation with negative affinity within the receptor pocket. Figure 1 shows the number of ligands

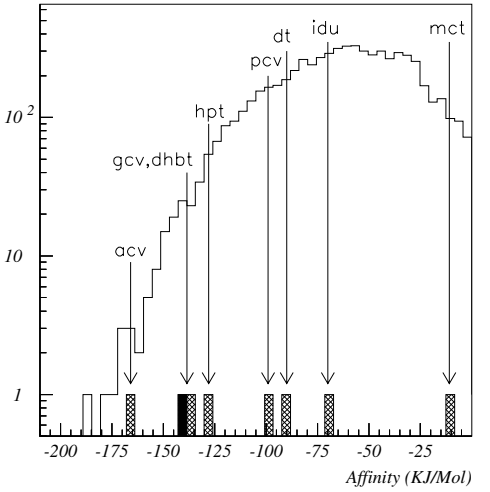


Fig. 1. Histogram of the affinities of the 6284 docked ligands (see text) to the rigid receptor conformation complexed with ganciclovir. The count is plotted on a logarithmic scale. While some known substrates, in particular the substrate corresponding to the receptor conformation, score well compared to the ligands of the randomly selected database, several other fail to achieve good scores.

Table 1. Ranking of the TK substrates in a screen of 10000 randomly chosen ligands of the nciopen3D database. The top row designates the crystal structure of the receptor that was used in the screen, the last column indicates the results of a flexible receptor screen.. (nd = not docked)

Substrate	1kim	1ki2	1ki3	1e2h	flex
acv	719	9	22	2048	199
ahiu	nd[c]	nd	nd	nd	2673
dhbt	4	104	118	38	13
dt	5	1310	2576	2779	681
gcv	3351	78	15	4516	57
hmtt	nd	nd	nd	nd	656
hpt	6	152	266	36	148
idu	515	2436	3272	2913	1365
mct	nd	6074	nd	nd	247
pcv	4845	952	4	4739	1656
Score:	3751	3705	4575	1926	4999

as a function of affinity and highlights the rank of the known TK substrates in the screen. Three structurally similar substrates, including the ligand associated with the receptor conformation, are ranked with very high affinity. This result demonstrates that docking method and scoring function are adequate to approximate the affinity of these ligands to the receptor. Four further ligands (idu, acv,gvc, pcv, for a detailed description of TK and its substrates we refer to [5]) docked badly, three further ligands did not dock at all according to the criteria above. Repeating the docking simulations for these ligands did not substantially improve their rank in the database, eliminating inaccuracies of the docking algorithm as the source for this difficulty.

The resulting ranks of this screen are summarized in Table 1 (second column), which displays the rankings of the 10 substrates. Three were ranked within the first 1%, 6 were ranked among the first 10% of the database, respectively. This enrichment rate is comparable to the results of other scoring functions that were previously investigated for this system, but the overall performance is disappointing [5].

Inspection of the crystal structures of the different receptor-ligand complexes reveal differences in the conformation of some side groups inside the receptor pocket, depending on the docked substrate. This is a well known fact, but it is often assumed that the impact of these conformational variations on the ranking accuracy is moderate. We therefore repeated the screening with the X-ray structure of TK in complex with the substrate gcv (ganciclovir, pdb entry: 1ki2 [16]), which had scored particularly bad in the original screen. The results are shown in Table 1 (third column). Now, gcv was ranked within the leading 1% of the database, but dt, formerly ranked on position 5, dropped to 1310. The same procedure was then repeatedw ith TK in complex with pcv (penciclovir, pdb entry: 1ki3), which raised its rank from 4845 to 4 (fourth column of Table 1).

For comparison purposes, we also performed a screen of the ligand free X-ray structure of TK (pdb entry: 1e2h [17]), which would most likely be used in a screen if no substrate was known. In this screen the receptor is unbiased to any of the substrates, which results in a dramatic loss of screening performance. As shown in column 7 only two ligands scored reasonably well (within the upper 10% of the database), all others would be discarded by any rational criterion as possible lead candidates.

3.2 Flexible Receptor Screens

Next we performed a flexible receptor screen against the same database. We identified the critical amino acid side chains and introduced 23 receptor degrees of freedom into the structure 1ki2, i.e. dihedral rotations of the amino acids His13(2), Gln76(3), Arg173(4), Glu176(4), Tyr52(3), Tyr123(3) and Glu34(4). The numbers in brackets indicate the degrees of freedom for each sidechain. Each step in the stochastic search now consisted of an additional random rotation for one receptor degree of freedom. The results of this screen are summarized in Fig. 2, the scores of the individual substrates listed in the column labeled ‘flex’ in the table. The figure demonstrates that in contrast to all rigid receptor screens

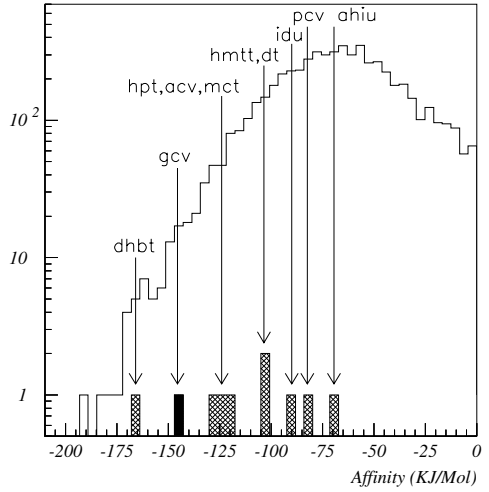


Fig. 2. Histogram of the affinities of the docked ligands in the flexible receptor screen.

now all substrates dock to the receptor. As expected, the number of false positives also increases, because a flexible conformation of the receptor reduces the bias of the screen against the known substrates. It must be noted that the accuracy of the flexible receptor screen is lower than that of the rigid receptor screens (with the same number of function evaluations) because the number of degrees of freedom has increased. The increased fluctuations in the flexible screen can be best seen for acv, where the optimization method failed to locate the global optimum of the affinity (as independently obtained in a longer scree for just this ligand). We are presently developing algorithms to only selectively move the sidechains to reduce the computational effort in the flexible receptor screen.

3.3 Comparison

To quantitatively compare different screens against the same ligand database, which used different receptor geometries, scoring functions or docking methods, it is sensible to assign an overall score to each screen which rates its performance [18]. We computed such a "score" for the entire screen from the ranks of the docked known substrates among the $N = 1000$ best ligands. This score is computed as the sum of $N - P$ where P is the rank of the known substrate and shown in the bottom row of table 1. A substrate ranking in the top of the screen contributes a score of 1000 to the sum, a badly ranked substrate comparatively little. Because the best N ligands are evaluated, screens which dock

many known substrates with moderate rank may have comparable scores with screens which perform perfectly for one substrate, but fail for all others. For the rigid receptor screens performed here the scores for the entire screen ranged from between 1926 for the screen against 1e2h, the ligand free X-ray structure of TK, to 4575 (1ki3, X-ray structure of TK in complex with pcv), which was arguably the best performing screen of all receptor conformations. Despite the increase in the number of false positives the overall score of the flexible receptor screen (4999) was better than that of any rigid receptor screens.

4 Conclusions

Our results offer a good demonstration that the ranking of known substrates can strongly depend on the particular receptor structure used for the screen. Binding energy and rank of a given substrate differ signigicantly depending on the receptor conformations. Our data demonstrate that this variability in rank is not, in general, a shortcoming of either scoring function nor docking methodology. Using a fixed three-dimensional structure of the receptor that is suitable for a single ligand introduces a significant bias in the overall scoring of the entire database. As a consequence, differences in the enrichment ratio for different scoring functions [5] may depend more on the suitability of the receptor conformation and environment than on the quality of the scoring function.

These findings suggest the importance of the consideration of a flexible binding pocket to obtain a better unbiased scoring of high-affinity ligands. The results of the flexible receptor screen reported here suggest that better accuracy of the scoring process can be achieved when receptor flexibility is considered. Ultimately only the routine use of accurate scoring techniques for flexible receptors, such as *FlexScreen*, will ameliorate this problem. The results presented here demonstrate that such screens will become feasible with present day computational resources in the near future.

Acknowledgments. We thank the Fond der Chemischen Industrie, the BMBF, the Deutsche Forschungsgemeinschaft (grant WE 1863/11-1) and the Kurt Eberhard Bode Stiftung for financial support.

References

1. Walters, W., Stahl, M., Murcko, M.: Virtual screening — an overview. Drug Discovery Today **3** (1998) 160
2. Drews, J.: Drug discovery: a historical perspective. Science **287** (2000) 1960
3. Klebe, G., Gohlke, H.: Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors. Angew. Chemie (Intl. Ed.) **41** (2002) 2644
4. Schneider, G., Boehm, H.: Virtual screening and fast automated docking methods. Drug Discovery Today **7** (2003) 64

5. Bissantz, C., Folkerts, G., Rognan, D.: Protein-based virtual screening of chemical databases. 1. evaluation of different docking/scoring combinations. J. Med. Chem. **43** (2000) 4759
6. Schnecke, V., Kuhn, L.: Virtual screening with solvation and ligand-induced complementarity. Persp. Drug. Des. Discovery **20** (2000) 171
7. Claußen, H., Buning, C., Rarey, M., Lengbauer, T.: Flexe: Efficient molecular docking consiudering protein structure variations. J. Mol. Biol. **308** (2001) 377–395
8. Osterberg, F.: Automated docking to multiple target structures: incorporation of protein mobility and structural water heterogeneity in autodock. Proteins **46** (2002) 34
9. Merlitz, H., Wenzel, W.: Comparison of stochastic optimization methods for receptor-ligand docking. Chem. Phys. Lett. **362** (2002) 271
10. Merlitz, H., Burghardt, B., Wenzel, W.: Application of the stochastic tunneling method to high throughput screening. Chem. Phys. Lett. **370** (2003) 68
11. Wenzel, W., Hamacher, K.: Stochastic tunneling approach for global optimization of complex potential energy landscapes. Phys. Rev. Lett. **82** (1999) 3003
12. Bolin, J., Filman, D., Matthews, A., Hamlin, R., Kraut, J.: Crystal structures of escherichia coli and lactobacillus casei dihydrofolate reductase refined at 1.7å resolution. J. Biol. Chem. **257** (1982) 13650
13. Jorgensen, W., McDonald, N.: Development of an all-atom force field for heterocycles.properties of liquid pyridine and diazenes. J. Mol. Struct. **424** (1997) 145
14. Morris, G., Goodsell, D., andR. Huey, R.H., Hart, W., Belew, R., Olson, A.: Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. J. Comput. Chem. **19** (1998) 1639
15. Milne, G., Nicklaus, M., Driscoll, J., S. Wang, a.D.Z.: National cancer institute drug information system 3d database. J. Chem. Inf. Comput. Sci. **34** (1994) 1219
16. Champness, J., Bennett, M., Wien, F., Visse, R., Summers, W., Herdewijn, P., de Clerq, E., Ostrowski, T., Sanderson, R.J.M.: Exploring the active site of herpes simplex virus type-1 thymidine kinase by x-ray crystallographyof complexes aciclovir and other ligands. Proteins **32** (1998) 350
17. Vogt, J., Perozzo, R., Pautsch, A., Prota, A., Schelling, P., Pilger, B., Folkerts, G., Scapozza, L., Schulz, G.: Nucleoside binding site of herpes simplex type 1 thymidine kinase analyzed by x-ray crystallography. Proteins **42** (2000)
18. Knegtel, R., Wagnet, M.: Efficiacy and selectivity in flexible database docking. Proteins **37** (1999)

A Sequence-Focused Parallelisation of EMBOSS on a Cluster of Workstations

Karl Podesta, Martin Crane, and Heather J. Ruskin

School of Computing, Dublin City University, Ireland
{kpodesta,mcrane,hruskin}@computing.dcu.ie

Abstract. A number of individual bioinformatics applications (particularly BLAST and other sequence searching methods) have recently been implemented over clusters of workstations to take advantage of extra processing power. Performance improvements are achieved for increasingly large sets of input data (sequences and databases), using these implementations. We present an analysis of programs in the EMBOSS suite based on increasing sequence size, and implement these programs in parallel over a cluster of workstations using sequence segmentation with overlap. We observe general increases in runtime for all programs, and examine the speedup for the most intensive ones to establish an optimum segmentation size for those programs across the cluster.

1 Introduction

Probably the most popular sequence analysis tool in present use is BLAST [1], a program for aligning sequences of biological data to investigate similarity. Single or multiple sequences form a query, which is searched against a database of sequences using a heuristic algorithm. The returned output is a list of matching target sequences. This whole process (a job) is repeated many thousands of times a day on publicly available BLAST servers. A number of parallel implementations [8,10,12,13,14] of the tool have been investigated and proposed at the levels of sequence, query, database, and job, to improve performance. Additionally, a number of other sequence analysis tools and implementations [3,6] (mostly alignment-based) have also been proposed which exploit parallelism at these levels.

In this work, we investigate if these methods can be applied to a wider range of tools (namely the EMBOSS suite [7]), and focus on the performance of the suite for increasing sizes of sequence. We present a parallel implementation that splits sequences into smaller ones (with an overlap to allow examination of the area at the split), and distributes these sub-sequences to a number of machines (or 'nodes') in a cluster of workstations.

The paper is organised into 3 major sections. The following section (Section 2) provides some background information on EMBOSS, along with a discussion of related work in parallel implementations of other programs. Section 3 presents a serial (non-parallel) analysis of programs in the EMBOSS suite. This is in order to examine the effect of increasing the size of input sequence data, and to identify potential opportunities for parallelisation, if any exist. A classification

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 473–480, 2004.
© Springer-Verlag Berlin Heidelberg 2004

of programs is discussed, along with methodology and results. Lastly, Section 4 presents a parallel implementation, and an examination of results for the more intensive programs in the EMBOSS suite. A final section presents our conclusions.

2 Background

2.1 EMBOSS

EMBOSS is a suite of programs (presently 160 programs as of release 2.8.0) used in molecular biology, whose primary function is to analyse sequences of biological data. The suite is freely available under the GNU Public License, is open source, and is maintained and developed at the Rosalind Franklin Centre for Genome Research (RFCGR) in Hinxton, Cambridge in the United Kingdom.

Programs are accessible through a number of methods (command line, web interface, and graphical user interface), and the suite is currently installed at, and supported by, a number of sites throughout EMBnet, the European network of molecular biologists, as well as a number of other sites around the world.

Although initially developed as a free replacement for a commercial software package, EMBOSS has a range of applications that offer a wide functional variety [4,15] capturing many common tasks of the bioinformatics specialist [9]. For example, programs in EMBOSS can be used to locally or globally align DNA sequences based on a number of techniques, predict the 2-dimensional structure of a protein, or calculate evolutionary distances between sequences in a multiple alignment. In addition, a wide number of data formats are supported and many public sequence databases (locally installed) can be used as sources of sequence data. Finally, there are also programs for displaying and editing sequences within the suite. A more complete classification of programs will be discussed in Section 3.2.

Further to the programs included in the suite is an Application Programmers Interface (API). This API provides core functions that can be used by programmers to create their own EMBOSS applications, and those applications of general benefit can be submitted for possible inclusion in the suite. Examples of core functions provided include those for reading and writing sequences in different formats, display functions, math functions, and string manipulation functions. All programs have a consistent interface, which allows programs to be embedded in batch runs, or attached to different interfaces with ease, thanks to the API. Importantly for this study, this facilitates a suite-wide analysis using common input sequences and parameters.

2.2 Related Work

Previous analyses of sequence analysis methods and programs have exploited parallelism at different levels, focused on decomposition of major components of the operation and on the operation itself. In [8], three levels of parallelisation within BLAST are proposed; fine grained (splitting a query), medium grained

(splitting a database), and coarse grained (splitting jobs). Only the last is further investigated, and uses the Portable Batch System (PBS) [2] to distribute BLAST jobs across a cluster of 25 CPUs. Other implementations that split BLAST jobs are detailed in [10] and [14], which use GNU Queue [16] and a custom Perl script to manage jobs, respectively.

At the database level of parallelisation, a database of sequences is split into a number of parts or 'fragments', and the same query is searched against each. In [12] and [13] for example, databases are divided into equal fragments on 20 compute nodes, with the analysis extended to many, smaller fragments to achieve better load balancing in the latter case. Additional overhead is incurred in formatting many fragments, which substantially decreases with additional processors, although an optimum value is not deduced.

At query level, the sub-sequences of a single query are examined. Work not based on BLAST is presented in [3] for homologous sequence retrieval, which uses a hybrid 'bucket' method to sort sequences and balance load. A different idea, related to job level parallelisation, is contained in [6], which uses a custom client/server to distribute queries to compute nodes.

Work presented at the level of a single sequence exclusively concerns genomes. In [11], splitting a single sequence into smaller sequences with overlap is contrasted with splitting a sequence into 'seeds' (which are identical to parts of the query sequence and extended to facilitate a similarity search).

In terms of sequence sizes, most investigations looked at average sizes of 1,600 base pairs (bp) [3] or less [8,10,12,13]. An exception [11] considered 940,000bp to be small (in the context of processing a genome), and also analysed medium and large sequences, the latter case constituting a full genome.

In all literature reviewed, examinations at the levels of sequence, query, database and job are exclusive. No hybrid combinations of methods have been implemented, with the exception of [6] and [11], with [11] comparing a hybrid method favourably with a single-level method. In addition, all implementations concern BLAST or similar sequence-search methods. EMBOSS in contrast, contains a wider variety of sequence analysis programs, and this paper seeks to determine if similar parallel methods can be applied across this spectrum.

3 Analysis of EMBOSS Programs

Parallel implementations of BLAST and similar programs are based on splitting data into smaller parts that can be processed in parallel across a number of processors. In order to determine if the same is applicable to programs in the EMBOSS suite, we have first examined the effects on program run time for increasing sizes of sequence data.

3.1 Methodology

EMBOSS programs were run 5 times each with the same sequence on a single machine, and the time averaged to eliminate atypical values. We repeated this

a number of times with a larger sequence each time, and plotted the results for execution time vs. size of sequence (Fig. 1). In order to show the effect of larger sequences, only the sequence size was changed, and default program parameters were used.

To implement this, a simple shell script was used that ran all programs in a batch job, recording times in a file. Each run of the batch file represented a single size of sequence, and needed to be run 5 times to average execution time, and a subsequent number of times for larger sequences. A number of programs took a different type of input, such as a nucleic acid sequence, protein sequence, or group of sequences, which were accounted for in the batch script.

Default parameters were defined within EMBOSS programs themselves, and in cases where they were not, example values from the EMBOSS documentation (distributed with the suite) were used. The sequence sizes we used started at 3,000bp (twice the size of the largest 'average' sized sequence used in work cited earlier), and were increased in increments of same, up to 15,000bp.

3.2 Classification

All 160 programs of the suite are classified into 33 groups, viewable either with the 'wossname' program, or on the EMBOSS website [15]. However, a number of programs are duplicated in more than one group, and a number of programs have similar functions (at least in the context of observing their response to larger data sizes). We have chosen to use the defined groupings regardless, in order to give a truer representation of effect across the function groups. However, for the purposes of brevity, we further classify the programs, based on group names in a defined list. These groups are Alignment, Display, Edit, Features, and Proteins.

A number of programs are eliminated from our examination on the basis that they are housekeeping utilities, or do not use either sequences or databases when run.

3.3 Results

The graphs in Fig. 1 detail run times (in log (seconds)) for programs with increasing sequence size (bp). A number of observations can immediately be made. Firstly, in each of the graphs there is a strong concentration of lines with the same trend, bar a few exceptions, telling us that relatively few programs show markedly extreme behaviour as sequence size increases. Secondly, the general trend for all programs is an increase in runtime. Thirdly, Fig. 1(a), Fig. 1(c), and Fig. 1(e) on the left hand side (Edit, Display, and Features respectively) vary in the space 0-1 seconds, whereas Fig. 1(b) and Fig. 1(d) on the right (Protein, and Alignment respectively) have mostly longer runtimes, from 0.1 seconds up to 654 seconds with 15,000bp. Clearly, non-housekeeping applications take longer (run-times greater than 1.0 seconds). A number of programs in particular have runtimes that are 10-100 times slower than the majority. The slowest 10 are (in ranked order) einverted, est2genome, matcher, dotmatcher, stretcher, wordmatch, dan, palindrome, sirna, and supermatcher, all described in [15].

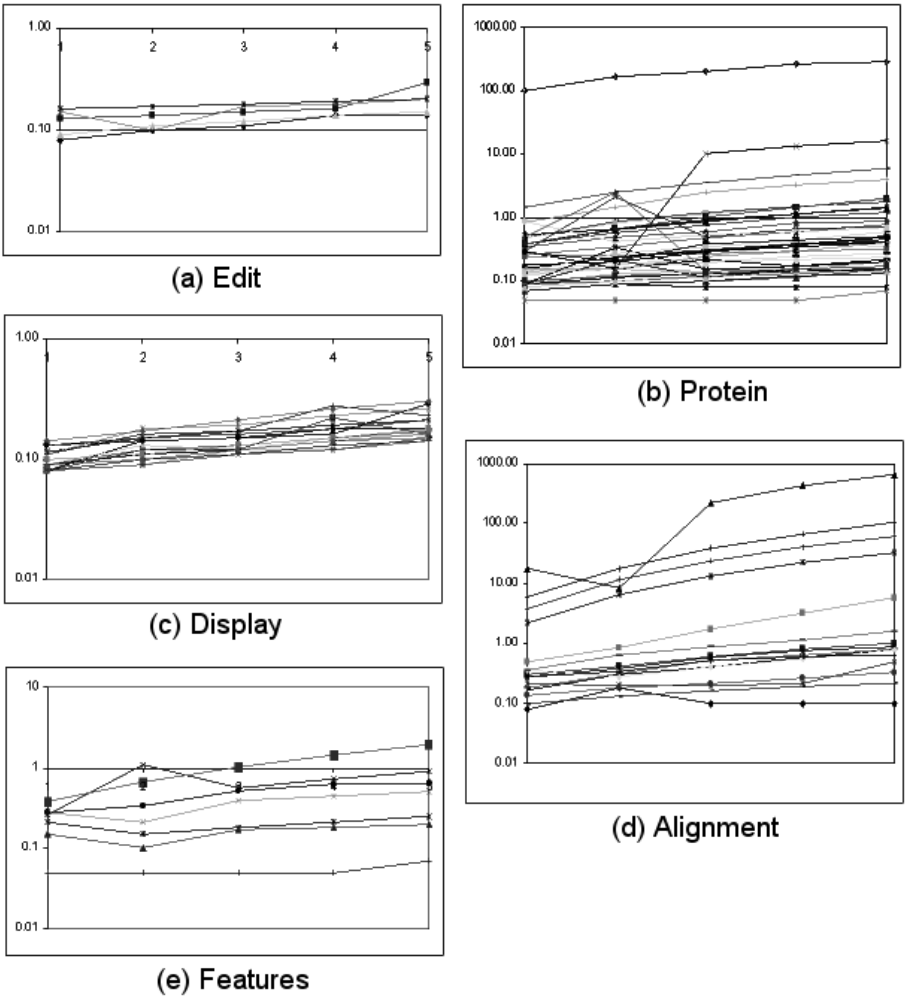


Fig. 1. Execution times for 110 programs of the EMBOSS suite, classified as per Section 3.2. X-axis values are in bp (3000, 6000, 9000, 12000, 15000), and Y-axis values are in log (seconds). The Protein and Alignment groupings contain the most intensive applications, with times in excess of 100 seconds for the largest sequence size

4 Parallelisation of EMBOSS

In this section we introduce a parallel implementation based solely on sequence segmentation, and examine the resulting performance across a cluster of workstations.

4.1 Methodology

For this implementation we adopted a method favoured by a number of previous implementations [10,14], namely the UNIX script wrapper. This method is fast and easy to develop, and allows program execution to be manipulated without alteration of source code (unlike an alternative method of implementation, MPI [5], which is an API for inserting message passing routines into source code). A disadvantage is that the scripts do not incorporate load-balancing capabilities available in [6], although those capabilities can be added. Strictly, these are not needed to evaluate the raw effect of a parallel implementation, which is of primary concern in this paper.

Using the EMBOSS API, the potential exists for sequences to be split within the suite by individual programs, although this places the burden of input segmentation on each individual program, while patterns may be common across a number of programs.

Using the example of [11], we split a query sequence into smaller sequences with an overlap. To do this we first use an EMBOSS program called 'splitter', which performs the split with a given overlap. Secondly, we remotely copy the smaller sequences to compute nodes using the UNIX utility "rcp". Thirdly, we initiate the analysis on the compute nodes using the UNIX utility "rsh", and finally, use rcp again to collate the results. The execution time required for this entire process is included in the results (Section 4.3), in order to show the scalability of all programs in the EMBOSS suite.

For the data, we chose a large sequence size (relative to the analysis in Section 3), i.e. 60,000bp. At four times the size of the largest data used in Section 3, this facilitates split sizes in a range that we have already examined. We keep this size constant to examine the improvements in performance over different numbers of nodes (the 'scalability'). The relative speedup tells us how fast the program performs compared to a serial run on a single node, and is defined simply as the time taken on n nodes divided by the serial time taken on a single node.

4.2 System

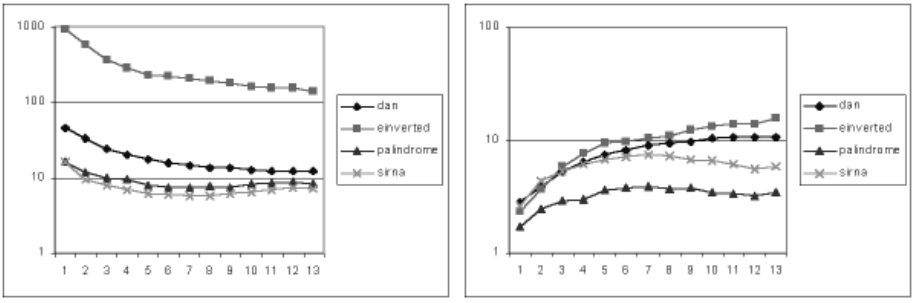
The system used is a modest lab-based cluster of 15 workstations, each with a single PentiumPro running at a clock speed of 180 megahertz (Mhz), 64 Megabytes (MB) of memory, and a 4 Gigabyte (GB) hard drive. Fast Ethernet at 100MB/sec connects the system.

4.3 Results

We illustrate results below on a small selection of programs, randomly taken from the group with the highest execution time (lowest performance). In both graphs (Fig. 2(a) detailing execution times and Fig. 2(b) detailing relative speedup) there is a sharp increase in performance for a small number of nodes, up to approximately 5, where there are minor increases until 12 nodes where performance

starts to decrease (e.g. palindrome and sirna). An optimal number of nodes to use can be taken as the lowest value from this interval.

It can be seen from the graphs that performance improvements degrade quicker for programs with shorter execution times, over a large number of nodes. For sirna (shortest execution times), speedup has changed into 'slowdown' over 13 nodes, whereas for einverted (highest execution times), a minor speedup is still maintained. Given that execution times increase for a single program given a larger sequence (as shown in Section 3), there is the potential for speedup to improve for any given program, given a larger sequence size (e.g. genome size).



(a) Execution Time (b) Relative Speedup

Fig. 2. Results for 4 programs most reactive to sequence size. X-axis values are number of nodes used. Y-axis values for (a) are execution times in log (seconds). Y-axis values for (b) are magnitude of speedup

5 Conclusions

In this paper we have presented both an analysis of 110 programs of the EMBOSS suite and a parallel implementation, based on the variable of sequence size. In the analysis of the suite, we were able to show that the runtimes of all programs were increased for an increasing sequence size. We also showed that the run times most affected were in the categories of Alignment and Proteins, whereas tools in Display, Edit, and Features, although affected, were substantially less so. With parallelisation based on sequence segmentation we were able to identify a plateau of optimum performance across a number of nodes (between 5 and 12, depending on the program used) for a selection of the most intensive applications, and suggest an improvement in program speedup given a larger size of sequence (as shown for programs with larger execution times). Overall, a parallelisation of EMBOSS programs is viable, at least from the point of view of increased sequence size, with major performance improvements possible for a small number of crucial suite applications in particular.

Acknowledgements. We would like to thank Dr. A. Bleasby and his group at the RFCGR in Hinxton, Cambridge for introducing the authors to EMBOSS. K. Podesta also acknowledges research support from the National Institute for Cellular Biotechnology (NICB), Ireland, under the HEA PRTLI scheme.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J.: A Basic Local Alignment Search Tool. Journal of Molecular Biology 215 (1990) 403–410
2. Feitelson, D.G., Rudolph, L. (Eds.): Job Scheduling Under the Portable Batch System. Lecture Notes in Computer Science, Vol. 949, Springer, Berlin (1995)
3. Yap, T.K., Frieder, O., and Martino, R.L.: Parallel Computation in Biological Sequence Analysis. IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 3 (1998)
4. Attwood, T.K., Parry-Smith, D.J.: Introduction to bioinformatics. Addison Wesley Longman (1999)
5. Gropp, W., Lusk, E., Skjellum, A.: Using MPI: Portable parallel programming with the Message Passing Interface. MIT Press (1999)
6. Clifford, R., and Mackey, A.J.: Disperse: a simple and efficient approach to parallel database searching. Bioinformatics, Vol. 16, No. 6 (2000) 564–565
7. Rice, P., Longden, I., Bleasby, A.: EMBOSS: The European Molecular Biology Open Software Suite. Trends in Genetics, Vol 16, Issue 6 (2000) 276–277
8. Braun, R.C., Pedretti, K.T., Casavant, T.L., Scheetz, T.E., Birkett, C.L., Roberts, C.A.: Parallelization of local BLAST service on workstation clusters. Future Generation Computer Systems, Vol. 17 (2001) 745–754
9. Stevens, R., Goble, C., Baker, P., Brass, A.: A classification of tasks in Bioinformatics. Bioinformatics, Vol. 17, No. 2 (2001) 745–755
10. Grant, J.D., Dunbrack, R.L., Mahon, F.J., Ochs, M.F.: BeoBLAST: distributed BLAST and PSI-BLAST on a Beowulf cluster. Bioinformatics, Vol. 18, No. 5 (2002) 765–766
11. Vincens, P., Badel-Chagnon, A., André, C., Hazout, S.: D-ASSIRC: distributed program for finding sequence similarities in genomes. Bioinformatics, Vol. 18, No. 3 (2002) 446–451
12. Mathog, D.R.: Parallel BLAST on split databases. Bioinformatics, Vol. 19, No. 14 (2003) 1865–1866
13. Darling, A.E., Carey, L., Feng, W.: The Design, Implementation, and Evaluation of mpiBLAST. ClusterWorld Conference & Expo and the 4th International Conference on Linux Clusters: The HPC Revolution (2003)
14. Hokamp, K., Shields, D.C., Wolfe, K.H., Caffrey, D.R.: Wrapping up BLAST and other applications for use on Unix clusters. Bioinformatics, Vol. 19 (2003) 441–442
15. http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Apps/groups.html. EMBOSS Application Groups. Accessed 13th January 2003
16. http://www.gnu.org/software/queue/queue.html. Queue load-balancing/distributed batch processing and local rsh replacement system. Accessed 13th January 2003

A Parallel Solution to Reverse Engineering Genetic Networks

Dorothy Bollman[1], Edusmildo Orozco[2], and Oscar Moreno[3]

[1] Department of Mathematics, UPRM
Mayagüez, Puerto Rico
bollman@cs.uprm.edu
[2] Doctoral Program in CISE, UPRM
Mayagüez, Puerto Rico
eorozco@cs.uprm.edu
[3] Department of Computer Science, UPR-Rio Piedras
Rio Piedras, Puerto Rico
O_MORENO@UPRI.UPR.CLU.EDU

Abstract. The reverse engineering problem is the problem of, given a set of measurements over a genetic network, determine the function that fits the data. In this work we develop a solution to this problem that uses a finite field model and that takes advantage of the efficient algorithms for finite field arithmetic that have recently been developed for other applications such as cryptography. Our solution, which is very efficient for the very large networks that biologists would like to consider, is given by a univariable polynomial which is determined by a parallel version of Lipson's interpolation algorithm.

1 Introduction

We are in the genome era. After decoding the human genome, the next stumbling block is to understand the function of genes and how they interact with each other, so that drugs can be created to cure diseases. Researchers in the area have proposed Boolean networks to describe the logic of the genes, in a manner similar to the way boolean functions describe the logic of computers. But it is useful to generalize the Boolean model to finite field models and thus take advantage of a number of efficient algorithms that have recently been developed for applications in error-correcting codes and public-key cryptography.

The reverse engineering problem can be described as follows: Given a set of biological measurements, determine the function that fits the data. Laubenbacher *et al* [7], [8] and Green [3] have addressed the reverse engineering problem for genetic networks using multivariable polynomials over finite fields and Groebner bases.

In this work we consider another approach which is computationally very efficient and will work for the very large networks that biologists want to consider in the near future. We consider a "lifting" method, described in more detail in Section 3, that consists of lifting a multivariable polynomial to a univariable

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 481–488, 2004.
© Springer-Verlag Berlin Heidelberg 2004

polynomial over a large finite field. There are very efficient algorithms available for the univariable case, as opposed to the Groebner basis methods for multivariable polynomials cited above.

2 Boolean and Finite Field Genetic Networks

Various researchers have described genetic regulatory networks using Boolean variables to represent gene expression levels or stimuli. For example, Ideker *et al* [6] give the following definition:
Definition. A *genetic network* consists of a directed graph having n numbered nodes such that for each node i there is an associated Boolean function f_i.

An expression matrix is a set of measurements (such as those which result from microarray experiments) over the genetic network. From this expression data, the challenge is to reconstruct or reverse engineer the genetic network.

In the Boolean model, either a gene can affect another gene or not. An alternative model that has been studied by several researchers [3], [7], [8], [10] is the finite field genetic network. In this model, one is able to capture graded differences in gene expression. Another advantage of the finite field model is that it can be considered as a generalization of the Boolean model since each Boolean operation can be expressed in terms of the sum and product in Z_2. In particular,

$$\begin{aligned} x \cap y &= x \cdot y \\ x \cup y &= x + y + x \cdot y \\ \tilde{x} &= x + 1 \end{aligned}$$

Considering a Boolean genetic network to be a finite field genetic network allows us to use the many tools for finite fields that have been developed for cryptology and communication theory.

It is very natural to consider a generalization of the Boolean model, not only to Z_2, but to an arbitrary finite field where the variables will now no longer be Boolean variables but now vary over $GF(p^m)$. To this end we define a *p^m-finite field variable* to be a variable over the finite field $GF(p^m)$. A *polynomial function* over $GF(p^m)$ is a multi-variable polynomial function in the variables $X_1, \ldots, X_n$ over the finite field $GF(p^m)$. Our new definition is then as follows
Definition. A finite field genetic network model is a set of n p^m-finite field variables (inputs), and a set of n polynomial functions over $GF(p^m)$ (outputs). As before the variables are the genes (or stimuli) and the polynomial functions give the gene network.

We clarify this with the following
Example: We define $f_0 = 1, f_1 = 1, f_2 = x_0 \cdot x_1, f_3 = x_1 \cdot (x_2 + 1)$. The only difference is that the variables now vary not over Z_2 but over the finite field with 4 elements GF(4). Note that GF(4) can be represented as: $GF(4) = \{0, 1, \alpha, \alpha^2\}$ where α is a root of the polynomial $z^2 + z + 1$ (with coefficients in Z_2). By taking $\{1,\ \alpha\}$ as a basis, we can say that $0 = (0,0)$, $1 = (0,1)$, $\alpha = (1,0)$, $\alpha^2 = (1,1)$.

Note in this example that we can naturally map the elements of the finite field (by taking a basis) into the integers from 0 to 2^{m-1} and in this manner we

can naturally model microarray data where now the variables can take all the values needed to accurately describe gradations in gene expression.

Genetic networks over finite fields allow many more variations in expression levels of the genes or the input stimuli, compared to the Boolean model which allows only two expression levels of two levels of stimuli

3 Reverse Engineering from Time-Series Data

The approach of Laubenbacher *et al* to solve the reverse engineering is, given a time series $S_1, S_2, \cdots, S_k$, where $S_i = (s_{i1}, s_{i2}, \cdots, s_{in})$, use Groebner bases to find a function f such that $f(S_1) = S_2, f(f(S_1)) = S_3, \cdots, f^{k-1}(S_1) = S_k$. Instead of finding a multivariable polynomial over $GF(p^m)$, we propose to find a univariable polynomial over $GF(p^{mn})$. This approach, which we call "lifting," is justified by the following theorem, which is proved in [10].
Theorem: For any fixed basis $\alpha_1, \alpha_2, \cdots, \alpha_n$ for $GF(p^m)^n$ there is a natural one-one correspondence between $(GF(p^m))^n$ and $GF(p^{mn})$.

Our solution to the reverse engineering problem thus consists of determining a univariable polynomial $P(x)$ such that

$$f(x) = P(x) + g(x),$$

where $P(x)$ is the univariable polynomial interpolating the points from the time series and $g(x)$ is a polynomial that vanishes at all interpolated values. Once having determined $P(x)$, one can use $g(x)$ to adjust the model. The polynomial $P(x)$ can be determined using the classical Lagrange interpolation formula, which has computational complexity $O(n^2)$, where n is the number of points to be interpolated. However, a faster polynomial interpolation algorithm is that of Lipson [9], which has computational complexity $O(n\ log^2\ n)$. Furthermore, as we shall see in Section 5, Lipson's algorithm can be readily parallelized. Our approach above is quite similar to algebraic coding, and it is very efficient when compared with the Groebner basis approach.

4 Fast Finite Field Arithmetic

In order to adapt either Lagrange or Lipson interpolation to finite fields, we need efficient algorithms for finite field arithmetic of polynomials over arbitrary finite fields. In the last few years there has been considerable progress in developing just such algorithms, particularly for applications in cryptography.

We assume that the coefficients of all polynomials over $GF(p^m)$ are written in the form α^i where α is a generator of the multiplicative cyclic group of $GF(p^m)$. The addition of two such polynomials then requires us to determine for a given a and b a number c such that $\alpha^c = \alpha^a + \alpha^b$. For the multiplication of two such polynomials we need to both add and multiply powers of α. This latter operation can be effected by simply adding the exponents modulo $p^m - 1$.

To add powers of α we use a table of Zech logarithms. Every element of $GF(p^m)$ can be written in the form $1+\alpha^i = \alpha^{z(i)}$ for some $z(i), 0 \leq z(i) \leq p^m-1$. We note that $\alpha^a + \alpha^b = \alpha^a(1 + \alpha^{(b-a) \ mod \ p^m-1})$. Hence, to add two powers of α we need only compute a Zech log and add exponents. It is useful to also note that, if p is odd, $-1 = \alpha^{(p^m-1)/2}$ and so $\alpha^a - \alpha^b$ can be computed by $\alpha^a + \alpha^{(p^m-1)/2} \cdot \alpha^b = \alpha^a + \alpha^{((p^m-1)/2+b) \mod p^m-1}$.

To construct a table of Zech logs, we first determine a primitive element α so that each field element x can be expressed as $x = \alpha^i$. We then construct an auxiliary table $A[i]$, $i = 1, 2, \cdots, p^m - 1$ such that each $A[i] = \alpha^i$. The table $Z[i]$ of Zech logs is then constructed by setting each $Z[i] = j$ where j is the index for which $A[j] = A[i] + 1$.

The following table gives the Zech logs for $GF(2^3)$ using the primitive polynomial $x^3 + x + 1$.

Table of Zech Logs for $GF(3^3)$.

i	1	2	3	4	5	6	7	8	9	10	11	12	13
$z(i)$	9	21	1	18	17	11	4	15	3	6	10	2	$*$

i	14	15	16	17	18	19	20	21	22	23	24	25	26
$z(i)$	16	25	22	20	7	23	5	12	14	24	19	8	13

The log table method is very efficient for small m. For large composite m, say $m = rs$, we have that $GF(2^m)$ is an extension of $GF(2^r)$ and we choose an irreducible polynomial of degree s over the "ground" field $GF(2^r)$. The numbers r and s should be chosen to both localize memory access for table lookup in the ground field as well as speeding up the mod reduction following the operation of polynomial multiplication.

5 A Parallel Interpolation Algorithm for Polynomials over Finite Fields

Lipson's algorithm [9] is based on the Chinese remainder theorem for polynomials, which says that given a set of n pairwise relatively prime polynomials $p_0(x),\ p_1(x),\ \ldots,\ p_{n-1}(x)$ and a set of residues $f_0(x),\ f_1(x),\ \ldots,\ f_{n-1}(x)$, there exists a unique polynomial $f(x)$ of degree less than the degree of $P(x) = p_0(x)p_1(x)\cdots p_{n-1}(x)$ which solves the set of congruences

$$f(x) = f_i(x) \mod p_i(x),\ i = 0, 1, \cdots, n-1$$

Polynomial $f(x)$ is given by:

$$f(x) = \sum_{i=0}^{n-1} e_i d_i f_i \mod P(x),$$

where $e_i = P'(x_i)$ (the formal derivative of $P(x)$ evaluated at $x = x_i$.)

In the special case of polynomial interpolation, the $p_i(x)$'s are of the form $p_i(x) = x - x_i$, which are relatively prime since the x_i are distinct.

Lipson's sequential interpolation algorithm is as follows:

Input: $\{(x_0, y_0), (x_1, y_1), \ldots, (x_{k-1}, y_{k-1})\}$, $k = 2^t$.

1. Compute $q_{i,j} = \prod_{m=i}^{i+2^j-1} p_m(x)$, $0 \leq j < t$, i a multiple of 2^j and $0 \leq i < k$.
2. Compute $d_i = (P'(x_i))^{-1}$, where P' is the derivative of $P(x) = q_{0,t} = (x - x_0)(x - x_1)\cdots(x - x_{n-1})$.
3. Compute $S_{i,0} = d_i * y_i$, for $0 \leq i < k$ and i a multiple of 2^j and $S_{i,j} = S_{i,j-1} * q_{i+2^{j-1},j-1} + S_{i+2^{j-1},j-1} * q_{i,j-1}$ for $0 < j < t$.

Output: The interpolating polynomial $f(x)$, $0 \leq deg(f(x)) < k$, such that $f(x_i) = y_i$.

It is easily shown that the arithmetic complexity of the above algorithm is $O(n\ log^2 n)$, which is superior to the $O(n^2)$ interpolation of Lagrange.

Let us examine the possibility of parallelizing Lipson's algorithm. We first note that $q_{i,j}$ and $S_{i,j}$ can be defined recursively:

$$q_{i,j} = q_{i,j-1} * q_{i+2^{j-1},j-1}$$

$$S_{i,j} = S_{i,j-1} * q_{i+2^{j-1},j-1} + S_{i+2^{j-1},j-1} * q_{i,j-1}$$

Thus, each of the $q_{i,j}$ and $S_{i,j}$ can be computed in parallel. The $S_{i,j}$ depend on the $q_{i,j}$ and in fact, the initial values $S_{i,0}$ depend on the values of $d_i = (P'(x_i))^{-1}$, which in turn depend on the last computed value of $q_{i,j}$, i.e., $q_{0,t}$. We note that $P'(x_i) = \prod_{0 \leq j \neq i \leq n-1}(x_i - x_j)$ and so the d_i can be computed independenly of $P(x)$ at no extra cost. We thus have following parallel version of Lipson:

for $(i = 0; i <= n; i++)$ **in parallel**
  $d_i = \prod_{0 \leq j \neq i \leq n-1}(x_i - x_j)$;
  $S_{i,0} = d_i * y_i$;
  $q_{i,0} = x - x_i$;
for $(j = 1; j < t; j++)$
  for $(i = 0; i < k; i += 2^j)$ **in parallel**
    $S_{i,j} = S_{i,j-1} * q_{i+2^{j-1},j-1} + S_{i+2^{j-1},j-1} * q_{i,j-1}$
    $q_{i,j} = q_{i,j-1} * q_{i+2^{j-1},j-1}$;

The computation can be represented by a binary tree with height $log\ k$. Load balancing is achieved by assigning a processor to every node. The only communications necessary are between parent nodes and their children.

Example

Let $\{(\alpha, \alpha^{16}),\ (\alpha^{16}, \alpha^{22}),\ (\alpha^{22}, \alpha^{15}),\ (\alpha^{15}, \alpha^2)\}$ be a set of four points in $GF(3^3) \times GF(3^3)$, where α is a primitive element of $GF(3^3)$.

• First, compute the d_i, $S_{i,0}$, and $q_{i,0}$.

$$\begin{aligned} d_0 &= [(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)]^{-1} \\ &= [(\alpha - \alpha^{16})(\alpha - \alpha^{22})(\alpha - \alpha^{15})]^{-1} \end{aligned}$$

$$
\begin{aligned}
&= (\alpha^{22}\alpha^{16}\alpha^{10})^{-1} \\
&= (\alpha^{22})^{-1} \\
&= \alpha^{4} \\
S_{0,0} &= \alpha^{4}\alpha^{16} \\
&= \alpha^{20} \\
q_{0,0} &= x - \alpha \\
d_1 &= [(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)]^{-1} \\
&= [(\alpha^{16} - \alpha)(\alpha^{16} - \alpha^{22})(\alpha^{16} - \alpha^{15})]^{-1} \\
&= (\alpha^{9}\alpha^{13}\alpha^{18})^{-1} \\
&= (\alpha^{14})^{-1} \\
&= \alpha^{12} \\
S_{1,0} &= \alpha^{12}\alpha^{22} \\
&= \alpha^{8} \\
q_{1,0} &= x - \alpha^{16} \\
d_2 &= [(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)]^{-1} \\
&= [(\alpha^{22} - \alpha)(\alpha^{22} - \alpha^{16})(\alpha^{22} - \alpha^{15})]^{-1} \\
&= (\alpha^{3}\alpha^{26}\alpha^{7})^{-1} \\
&= (\alpha^{10})^{-1} = \alpha^{16} \\
S_{2,0} &= \alpha^{16}\alpha^{15} \\
&= \alpha^{5} \\
q_{2,0} &= x - \alpha^{22} \\
d_3 &= [(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)]^{-1} \\
&= [(\alpha^{15} - \alpha)(\alpha^{15} - \alpha^{16})(\alpha^{15} - \alpha^{2}]^{-1} \\
&= (\alpha^{23}\alpha^{5}\alpha^{20})^{-1} \\
&= (\alpha^{22})^{-1} = \alpha^{4} \\
S_{3,0} &= \alpha^{4}\alpha^{2} \\
&= \alpha^{6} \\
q_{3,0} &= x - \alpha^{15}
\end{aligned}
$$

• Second, compute the $S_{i,1}$ and $q_{i,1}$

$$
\begin{aligned}
S_{0,1} &= S_{0,0} * q_{1,0} + S_{1,0} * q_{0,0} \\
&= \alpha^{20} * (x - \alpha^{16}) + \alpha^{8}(x - \alpha) \\
&= \alpha^{10}x + \alpha^{5} \\
S_{2,1} &= S_{2,0} * q_{3,0} + S_{3,0} * q_{2,0} \\
&= \alpha^{14}x + \alpha^{22} \\
&= \alpha^{5} * (x - \alpha^{15}) + \alpha^{6} * (x - \alpha^{22}) \\
q_{0,1} &= q_{0,0} * q_{1,0}
\end{aligned}
$$

$$
\begin{aligned}
&= (x-\alpha)(x-\alpha^{16}) \\
&= x^2 + \alpha^{13}x + \alpha^{17} \\
q_{2,1} &= q_{2,0} * q_{3,0} \\
&= (x-\alpha^{22})(x-\alpha^{15}) \\
&= x^2 + \alpha^{6}x + \alpha^{11}
\end{aligned}
$$

- Third, compute $S_{0,2}$

$$
\begin{aligned}
S_{0,2} &= S_{0,1} * q_{2,1} + S_{2,1} * q_{0,1} \\
&= (\alpha^{10}x + \alpha^5) * (x^2 + \alpha^6 x + \alpha^{11}) + (\alpha^{14}x + \alpha^{22}) * (x^2 + \alpha^{13}x + \alpha^{17}) \\
&= x^4 + \alpha^{10}x^3 + \alpha^{20}x^2 + \alpha^6 x + \alpha^2
\end{aligned}
$$

Thus the polynomial that interpolates the four given points is

$$f(x) = x^4 + \alpha^{10}x^3 + \alpha^{20}x^2 + \alpha^6 x + \alpha^2.$$

6 Conclusions and Future Work

The finite field model for genetic networks has several advantages over the Boolean model. It is more general than the Boolean model and allows one to capture graded differences in gene expression. It also allows us to take advantage of a number of efficient algorithms for finite field arithmetic. The liftng method allows us to describe the solution of the reverse engineering problem in terms of a univariable polynomial. and for this, we have adapted a parallel version of Lipson's polynomial interpolation algorithm. Our method is very efficient for very large genetic networks, as opposed to other known Groebner base methods for multivariable polynomials.

We are presently developing software in C for finite field arithmetic for characteristic two and a C/MPI program for Lipson's algorithm. Future work contemplates the development of table lookup methods for finite fields of charteristic $p \neq 2$.

References

1. T. Akutsu and S. Kuahara and O. Maruyama and S. Miyano: "Identification of Gene Regulatory Networks by Strategic Gene Disruptions and Gene Overexpressions", Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms, H. Karloff, ACM Press, 1998
2. E. DeWin, A. Bosselaers, S. Vandenberghe, P. DeGersem, and J. Vandewalle: "A fast software implementation for arithmetic operations in $GF(2^n)$," In K. Kim and T. Matsumoto, editors, *Advances in Cryptology - ASIACRYPT 96*, Lecture Notes in Computer Science, No. 1163, pp. 65–76, Springer-Verlag, Berlin, 1999
3. E. L. Green: "On polynomial solutions to reverse engineering problems,"preprint

4. J. Guajardo and C. Paar: "Efficient algorithms for elliptic curve cryptosystems." In B.S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO 97,* Lecture Notes in Computer Science, No. 1294, pp. 342–356, Springer-Verlag, Berlin, 1997
5. M. Anwarul Hasan: "Look-up table-based large finite field multiplication in memory constrained cryptosystes," IEEE Trans. Computing,Vol. 49, No.7, pp. 749–758, 2000
6. T.E. Ideker, V. Thorsson, and R.M. Karp: "Discovery of regulatory interactions through perturbation: Inference and experimental design," Pacific Symposium on Biocomputing, No. 5 pp. 302–313, 2000.
7. R. Laubenbacher and B. Stigler: "Dynamic networks," Adv. in Appl. Math Vol.26, 2001, pp. 237–251
8. R. Laubenbacher and B. Stigler: "Biochemical networks,"preprint
9. J. Lipson: "Chinese remaindering and interpolation algorithms," Proc. 2nd Symposium in Symbolic and Algebraic Manipulation," pp. 372–391, 1971
10. O. Moreno, D. Bollman, and M. Aviño: "Finite dynamical systems, linear automata, and finite fields," 2002 WSEAS Int. Conf. on System Science, Applied Mathematics & Computer Science and Power Engineering Systems, pp. 1481–1483. Also to appear in the International Journal for Computer Research
11. H. Ortiz, M.A. Aviño, S. Peña, R. Laubenbacher, O. Moreno: "Finite Fields are Better Boolean, ," Proc. of the Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB), Vol. 2003, pp. 162
12. E. Savas and C.K. Koc: "Efficient methods for composite field arithmetic," Technical Report, Oregon State University, 1999
13. B. Sunar, E. Savas, and C.K. Koc: "Constructing Field Representations for efficient conversion," to appear in IEEE Transactions on Computers

Deformable Templates for Recognizing the Shape of the Zebra Fish Egg Cell

Ho-Dong Lee[1], Min-Soo Jang[1], Seok-Joo Lee[1], Yong-Guk Kim[2], Byungkyu Kim[3], and Gwi-Tae Park[1]

[1] Dept. of Electrical Engineering, Korea University, Seoul, Korea
gtpark@elec.korea.ac.kr
[2] School of Computer Engineering, Sejong University, Seoul, Korea
ykim@sejong.ac.kr
[3] Korea Institute of Science and Technology, Seoul, Korea

Abstract. In biology, manipulating a micro-scale object such as chromosome, nucleus or embryo has been an important issue. For instance, skillful manipulation of the embryo cell in the biological experiment requires many years experience with a complex setup. Moreover, such process is usually very slow and requires many hours of intense operations such as trying to find the position of the cell within a petri dish and injecting a pipette to the cell from the best orientation. We have designed a new vision system, by which it finds the region of the zebra fish egg cell, and then tracks the yolk as well as the inner line that divides between protoplasm and vitellin within the egg cell, using two different deformable templates. Result suggests that recognition performance of the system varies depending upon the magnification ratio of the microscope, and it reaches to the maximum at 80:1.

1 Introduction

Manipulating an object within the micro scale is a key technology in biology, since the sizes of DNA, chromosome, the nucleus, the cell and embryo are within the order of micrometer [2], [3], [5], [8]. However, such manipulation is typically difficult, because the objects are weak, small (e.g. the size of the zebra fish egg cell is about 1mm – 2mm), and the operation should be carried out in the slippery culture fluid. And yet, most of such micromanipulations are carried out manually. Therefore, the manipulators should often spend over a year to accomplish an experiment. Since they depend on visual inspection, the success rate and efficiency of manipulation is extremely low because of eyestrain. For such reasons, automation of the micromanipulation has been asked.

In the zebra fish egg cell manipulation, the insertion position of the pipette is determined by the structure of the egg cell. In this paper, we propose a new scheme where the machine finds the zebra fish egg cell and then recognizes inner structures of the cell, such as yolk and the inner line that divides between protoplasm and vitellin using the deformable templates. Because the egg cell is alive and laying in the culture fluid, the conventional vision algorithms such as template matching and mathematical morphology operation show some limitation in recognizing the structure of the cells [1], [7]. Moreover, the several factors such as diverse size and shape of living cells

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 489–497, 2004.
© Springer-Verlag Berlin Heidelberg 2004

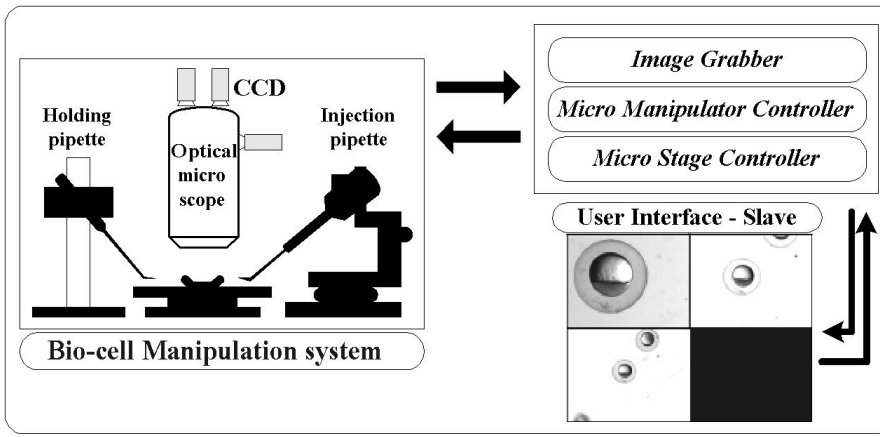


Fig. 1. A schematic view of the micromanipulation system

and optical characteristic of the culture fluid make it difficult to recognize the cell accurately. It is found that the deformable template method is the better choice [9]. We define first two deformable templates for the yolk and inner line for an egg cell, respectively. And then, the energy function for each template measures whether it is best fit or not. The deformable template and the energy function allow us to recognize accurate position of inner structure of the egg cell.

For the micromanipulation, an optical microscope is typically used. In our setup, we utilize three images, acquired from the microscope having the same viewpoint with three different magnification ratios. The low magnification image is used for searching the ROI (Region Of Interest), whereas the high magnification image for recognizing the inner structure of the cell. Utilization of multiple images increases the efficiency and the precision. In section 2, the details of our micromanipulation system are described. The proposed algorithm for recognizing the zebra fish egg cell is discussed in section 3 and section 4. Result of experiments is described in section 5. Finally, we summarize our results, and discuss the performance of the whole system in section 6.

2 The Micromanipulation System

The present vision-based micro-manipulation system consists of three main parts: first, an optical stereomicroscope with three CCD cameras mounted on top of the microscope; secondly, the micro XY stage and micromanipulator for a holding pipette and an injection pipette; thirdly, a PC with a controller for the micro XY stage, the micromanipulator and the image grabber. Fig. 1 shows the vision based micro-manipulation system and its schematic architecture.

The system contains an optical stereomicroscope with three magnification ratios (e.g. x240, x120 and x80). So, three images can be acquired simultaneously. The micro XY stage is driven by lead-screws for translation movement with a travel range of 25mm. Precision roller bearings guarantee straightness of travel within 2μm. It uses a compact closed-loop DC motor with a shaft-mounted high-resolution position

encoder, and the precision gear provides 0.1μm minimum incremental motion with a resolution of 0.0085μm. Both the injection pipette and holding pipette are installed on the micromanipulator with 3DOF (Degree Of Freedom), actuated by three stepping motors. The mobile range of each axis is 25.4mm and the step resolution of the stepping motor is 0.04μm.

3 The Operation of the System

In this section, we describe how the system is operated, and the merit of using the multiple magnification images. By nature, as the magnified image provides more precise view, we can have better information of the target. However, the highly magnified image brings also a narrower visual field to the viewer. Moreover, any slight movement of holding or injection pipette conveys a large displacement in such image. On the other hand, since the less magnified image shows a wide area, one can find the egg cell within the image rapidly, and ignore small amount of displacement error. However, the lower resolution of such image makes it difficult to recognize exact position of inner structures of the egg cell. Fig. 2(a) shows three different cases within the user interface. These three windows show images that have different magnification ratios. As you can see, the highly magnified one provides a more detail information about inner structure of the egg cell than the less magnified one.

The use of those multiple images increases efficiency and precision of the micromanipulation system. Initially, the system extracts the ROI within a given image with a lower magnification ratio by using histogram segmentation algorithm as described in section 4. When the system has succeeded in extracting the ROI area of the cell, it generates a moving trajectory to the center of the image by moving the micro XY stage using visual feedback. Then, it starts to search about the inner structures of the egg cell. However, when it fails, it regards that cell as an irregular egg cell. Fig. 2(b) depicts the flowchart of the micromanipulation system.

Our target cell here is the zebra fish egg cell. It is widely used for the biological experiments, because it is a vertebrata like human being and it takes only two days for

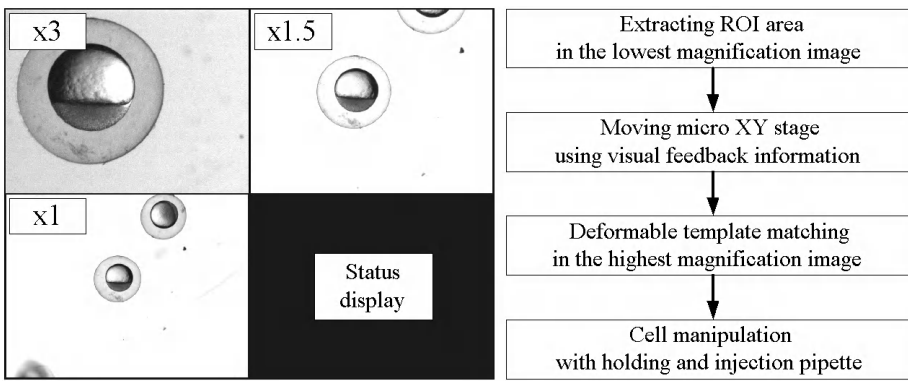


Fig. 2. The user interface (a) and the flow chart (b) of the system

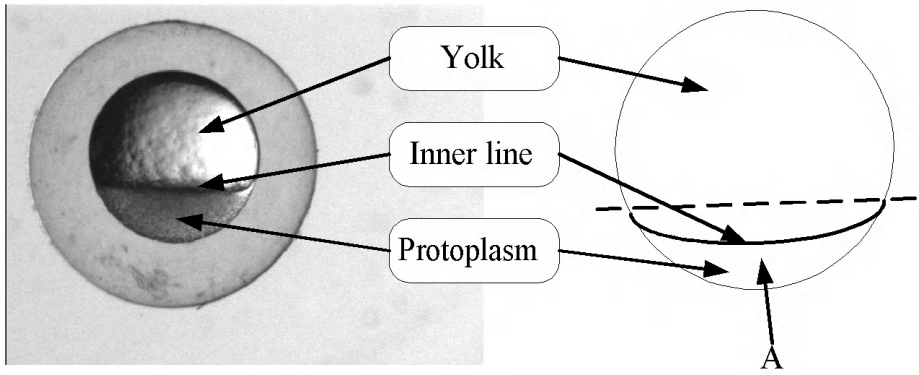


Fig. 3. Structure of the zebra fish egg cell

hatching. Fig. 3 shows the typical structure of a zebra fish egg cell, consisting of yolk, protoplasm and an inner line. Since we want to prevent any chance of destruction of the egg cell during insertion of the injection pipette into the protoplasm, it is essential to recognize the exact positions of the inner line. Notice that the arrow A in Fig. 3 indicates the potential injecting direction of the pipette, which is perpendicular to the dotted line drawn over the end points of the inner line.

4 Shape Recognition of the Zebra Fish Egg Cell

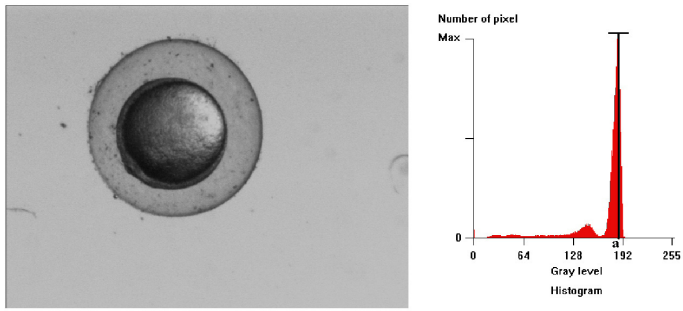
In this section, we describe a recognition algorithm for the zebra fish egg cells. The present vision algorithm consists of two main parts: the ROI extraction and recognition of the egg cell.

4.1 Extraction of the ROI

To recognize the structure of the egg cell, we need to extract the ROI from the acquired image. This process increases the recognition rate of the system and reduces the computational time, because we can apply the deformable template method only to the ROI selectively. The process consists of histogram segmentation and the nearest neighborhood method [4]. Fig. 4(a) shows the zebra fish egg cells embedded on the background, and Fig. 4(b) is a histogram for the same image. Since the background area occupies the majority of the image, the peak in the histogram corresponds to the background area. By using this characteristic, we can easily eliminate the background area.

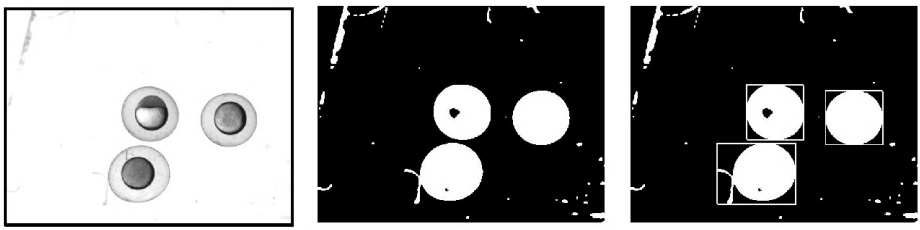
Fig. 5 illustrates three steps how the ROI can be extracted from the given image. In Fig. 5(a), the background gray pixels are removed from the image. After digitization of this image, each egg cell area becomes a group of white pixels as shown in Fig. 5(b).

Then the nearest neighbor algorithm finds the nearest pair of distinct clusters and merges the pair of clusters. The result image is shown in Fig. 5(c), in which three distinctive clusters are emerged.



(a) (b)

Fig. 4. Histogram of the cell image



(a) (b) (c)

Fig. 5. Steps of determination of the ROI

4.2 Deformable Template Model

In general, the deformable template method models a target object using a template with a few parameters [6], [9]. It aims to recognize the target object by adjusting the parameters. Although one can have a complex deformable template using many parameters, then the search time will increase rapidly. Therefore the complexity of a deformable template model must trade off between the search time and the recognition accuracy. To simplify our task, we made two assumptions:

(a) Shape of the yolk is similar to a circle.
(b) The inner line is modeling with a Bezier curve [10], [11].

4.2.1 Deformable Template for the Yolk

Fig. 6(a) is the deformable template for the yolk of the zebra fish egg cell. It has three parameters, x, y and r, corresponding to two coordinates for the center and the radius of the template, respectively. Three parameters are selected for modeling the yolk according to the assumptions (a). An arbitrary point on the template is defined as $p(x, y)$. From that point, the upper and lower contours are given by

$$\begin{aligned} &\textit{Upper contour } p(x,y) = \left\{ x \in [-r, r] \,|\, y = \sqrt{r^2 - x^2} \right\} \\ &\textit{Lower contour } p(x,y) = \left\{ x \in [-r, r] \,|\, y = -\sqrt{r^2 - x^2} \right\} \end{aligned} \tag{1}$$

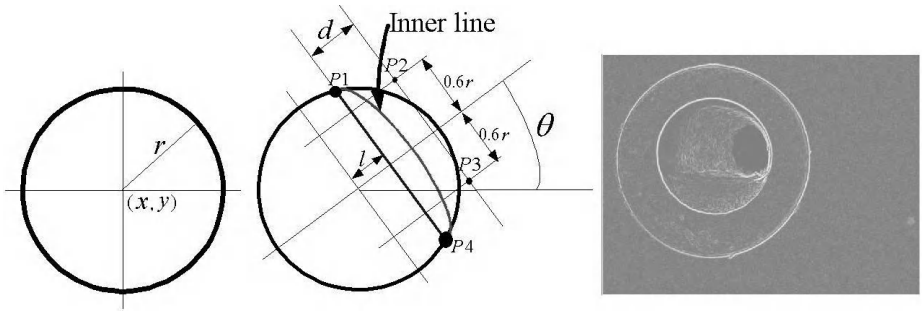


Fig. 6. Two deformable templates and the potential field

4.2.2 Deformable Template for the Inner Line

Fig. 6(b) is the deformable template for the inner line located within a zebra fish egg cell. To model this template, we need four points, *P*1, *P*2, *P*3 and *P*4, for drawing a Bezier curve. And we can introduce three parameters *l, d* and θ, corresponding to a distance between center of the circle to the line $\overline{P1P4}$, a distance between the line $\overline{P1P4}$ to the line $\overline{P2P3}$, and the rotational degree of the template, respectively. The relationship between four points on the template and three parameters are defined as follow:

$$\begin{aligned} &P1(x^{'}, y^{'}) : x^{'} = x + l,\ y^{'} = y - \sqrt{r^2 - l^2} \\ &P2(x^{'}, y^{'}) : x^{'} = x + l + d,\ y^{'} = y - 0.6r \\ &P3(x^{'}, y^{'}) : x^{'} = x + l + d,\ y^{'} = y + 0.6r \\ &P4(x^{'}, y^{'}) : x^{'} = x + l,\ y^{'} = y + \sqrt{r^2 - l^2} \end{aligned} \tag{2}$$

where 0.6*r* defined heuristic approach to all points on the Bezier curve into the circle. And an arbitrary point where is rotated by θ on the template is defined as follow:

$$p(x^{'}, y^{'}) \ :\ \mathrm{x}^{'} = x\cos\theta + y\sin\theta,\ \mathrm{y}^{'} = -x\sin\theta + y\cos\theta \tag{3}$$

4.2.3 Energy Function

To determine whether the ROI is a yolk or not, an energy function is introduced. The energy function is defined by equation (4). Here, as the template models the yolk accurately, the energy function becomes larger.

$$E_{total} = E_{edge} + E_{deviation} \tag{4}$$

The energy function consists of two terms: the edge enhancement term E_{edge} and the deviation term $E_{deviation}$. The first term is the summation of the potential fields for the coordinates on the template contour,

$$E_{edge} = \frac{1}{255n}\sum_{i=0}^{n}\phi_{edge}^{i}(x,y) \tag{5}$$

where n is the number of pixels on the template contour and the potential field ϕ_{edge} is the edge enhanced image by using Sobel operator [4]. The potential field ϕ_{edge} of an image of the zebra fish egg cells is shown in Fig. 6(c), where one can observe a lot of noise even after applying the noise elimination filter. Since such noise decreases the recognition rate, a deviation term is adopted. The deviation term $E_{deviation}$ is given by

$$E_{deviation} = 1 - \frac{1}{255}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(ave - \phi_{edge}^{i}(x,y))^{2}} \tag{6}$$

where ave is an average of the field potentials on the coordinates of the template contour. From the equation (6), we know that the deviation term is decreased when the edge potentials are deviated from the given average value, whereas it approaches to the maximum value 1 as the average value is equal to the field potentials. By using the deviation term, we can increase the recognition rate and stability of the energy function.

4.3 Application of Deformable Templates to the Zebra Fish Egg Cell

Once an ROI is selected as shown in Fig. 5(c), the system starts to find the yolk within the region, by scanning from the upper-left to the bottom-right, drawn as a gray rectangle in Fig. 7(a). The shape of the template is, of course, a circle since it is looking for a yolk of the zebra fish egg cell. The parameters for this template are two coordinates for the center and a radius of the circle. The criterion for decision is based upon the total value of the energy function.

Following the recognition of the yolk, the next step is to find an inner line in the area of the yolk as illustrated in Fig. 7(b), since it is assumed that the inner line is within the yolk. In this case, the scanning is carried out similar to the above case in the yolk area.

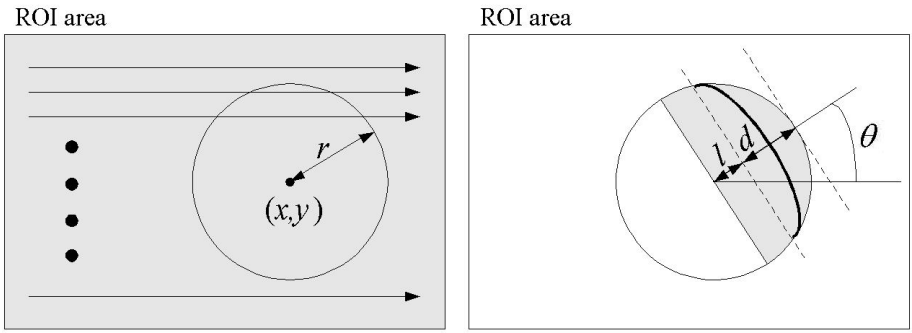
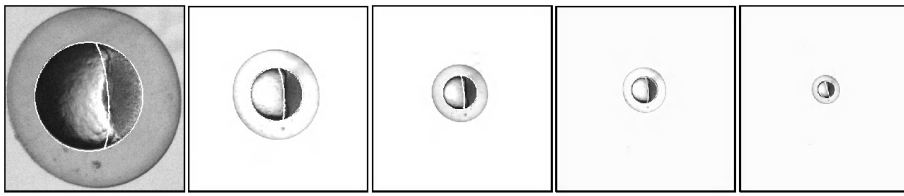


Fig. 7. Application of deformable template to the yolk (a) and the inner line (b) within the ROI



(a) (b) (c) (d) (e)

Fig. 8. Superposition of deformable templates to the zebra fish egg cell images

Table 1. Processing speeds and recognition rates for different magnification ratios

Magnification ratio	Processing time	Recognition rate
240:1	5.117sec	71%
120:1	0.795sec	77%
80:1	0.502sec	97%
60:1	0.449sec	80%
40:1	0.442sec	2%

5 Experiment and Performance of the System

We construct a zebra fish egg cell image database to evaluate the performance of the proposed zebra fish egg cell recognition algorithm. All images in the present database are acquired from the micromanipulation system described in section 2, and the database contains 880 zebra fish egg cell images with 240:1, 120:1, 80:1, 60:1, and 40:1 magnification ratio, respectively. Fig. 8 shows 5 image shots for a zebra fish egg cell, and two deformable templates (i.e. the circular one for the yolk and the curve in the circle for the inner line, respectively) are superimposed on each image.

Table 1 summarizes our results for five magnification cases. It shows that the lower the magnification ratio, the faster the processing time for each cell. The shape recognition rate was initially increased as the magnification ratio was decreased. But the recognition rate was decreased as the magnification ratio becomes lower than 80:1. In fact, even the human observer has some difficulty in recognizing the inner line for such cases. Notice that we counted it as a success case only if our algorithm recognized both the yolk and the inner line. The processing time is about 0.5s at the 80:1 ratio. The moving time scale of other mechanical part such as injection pipette, holding pipette, and micro stage are relatively slower than the processing time.

6 Conclusions and Discussion

To develop an automatic micromanipulation system for the zebra fish egg cells, we designed a vision system where a new deformable template algorithm was incorporated with the multiple view operation for recognizing the shape of the zebra fish egg cell. We have modified the deformable template method for the present task:

one for the circular-shaped yolk and the other for the curve-shaped inner line of the zebra fish egg cell. For the stability and performance of the method, the energy function includes deviation term as well as the edge potential field term. For the preprocessing stage, we used the histogram segmentation for extracting the cell body area from the background. And the dynamic search area algorithm was developed to reduce the processing time. Result suggests that there is an optimum magnification ratio where the recognition rate approaches to the maximum. The present study demonstrated that deformable template is useful in recognizing the inner structure of the micro-scale cell.

References

1. D. Anoraganingrum, "Cell Segmentation with median filter and mathematical morphology Operation", Proc of International Conference on Image Analysis and Processing, pp.1043–1046, 1999
2. F. Arai, A. Kawaji, P. Luangjarmekorn, T. Fukuda and K. Itogawa, "Three-dimensional bio-micromanipulation under the microscope", Proc. of IEEE ICRA, pp. 604–609, 2001.
3. F. Arai, T. Sugiyama, P. Luangjarmekorn, A. Kawaji, T. Fukuda, K. Itoigawa, and A. Maeda, "3D bio-micromanipulation", International Symposium on Micromechatronics and Human Science, pp. 71–77, 1999
4. G. A. Baxes, "Digital Image Processing: Principles and Applications", John Wiley & Sons, Inc. 1994.
5. Y. Kimura, R. Yanagimachi, "Intracytoplasmic Sperm Injection in the Mouse", Biology of Reproduction, Vol. 52, No.4, pp. 709–720, 1995
6. G-S. Lee, "Automatic Face Region Detection Using Chromaticity Space and Deformable Template", Master thesis, Korea University, 2001.
7. X. Li, G. Zong and S. Bi, "Development of global vision system for biological automatic micromanipulation system", Proc. of IEEE ICRA, pp. 127–132, 2001.
8. S. Yu, and B. J. Nelson, "Microrobotics cell injection", Proc. of the IEEE ICRA, pp. 620–625, 2001
9. A. Yuille, D. S. Cohen and P. W. Hallinan, "Feature extraction from faces using deformable templates", CVPR'89, pp. 104–109, 1989
10. http://www.moshplant.com/direct-or/bezier/
11. http://astronomy.swin.edu.au/~pbourke/curves/bezier/

Multiple Parameterisation of Human Immune Response in HIV: Many-Cell Models

Yu Feng, Heather J. Ruskin, and Yongle Liu

School of Computing,
Dublin City University,
Dublin 9, Ireland
{fengyu,hruskin,yliu}@computing.dcu.ie
http://www.dcu.ie/computing/msc/index.html

Abstract. Mathematical and computational models of the Human Immune Response have gained considerable attention over recent years and a number of approaches have been reported in the literature. One of the most successful relies on modelling, at cell level, the key components of the response using cellular automata/Monte Carlo strategies. However, a core issue remains the parameterisation required to demonstrate realistic evolution. We discuss a model of 8 cell-types, which can represent both T cell-mediated and humoral functions of the immune system, and focus on parameter sets, with values chosen to reflect realistic time-scales, comparable to natural biological processes. Analysis of the influence of the parameters introduced enables comparison of the properties of the 8-cell and basic models. In particular, a slightly reduced critical mutation value is found to lead to immune deficiency while, when a variable mutation growth factor is applied, immune breakdown occurs rapidly. The 8-cell model is susceptible to some reduction and aggregation, but system "fitness" dominates response.

1 Introduction

Numerous attempts have been made in theoretical immunology to understand the population dynamics of cells and the way in which the growth and decay of these cellular elements is balanced and co-ordinated to mount an effective response. One main approach is to describe the continuous evolution of the cellular elements as a set of differential equations, [1], the other to allow the cell populations to evolve iteratively through a simplified set of rules and interactions applied discretely to each cell-type, [2]. Some microscopic models have been predominantly concerned with a comprehensive picture of the behaviour of immune system components and their interactions, [3]-[8], while others, usually involving fewer entities and simplification of the inter- and intra-cellular interactions, have concentrated on charting the overall disease evolution, [9]- [13]. A key issue for the latter is the crossover in immune status, as demonstrated by the change in population levels of the various cell-types. While most viruses are eventually suppressed by the immune system, the human immuno-deficiency

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 498–507, 2004.
© Springer-Verlag Berlin Heidelberg 2004

virus, (HIV), in acquired immune deficiency syndrome, (AIDS), is a retrovirus with a unique ability to evade destruction. For HIV invaders, the main focus is destruction of helper T4-cells in the immune system, but they also infect and replicate in macrophages, mucous membranes, the liver, spleen, brain and lymph nodes amongst others, leading to serious impairment of function, if not ultimate destruction of the cell. Cell characteristics, such as the potential of the virus to mutate, the mobility of viral and host cells, transfer rates between cell-types (e.g. B-cells to memory B-cells) and so on, affect the fitness of the whole system and its ability to respond to attack. Such characteristics may be modelled by a set of stochastic parameters, which may be fixed for the whole evolutionary cycle, ([13] and references), or may change according to changes in the viral growth pattern, [14]. In recent work, an attempt was made to bridge the gap between comprehensive microscopic detail, such as that described through IMMSIM, [4], [7], and key global effects, [9]-[13], by investigating the behaviour obtained from a so-called "mesoscopic" model [15], which incorporated some elements of humoral, as well as cell-mediated response. For the primary and secondary reactions, T-cell polarisation was considered, together with the implications for persistence of immunological memory. This attempt to explain inter-dependencies in the response also motivates, to some extent, recent efforts on continuum and hybrid models of the immune repertoire, formulated wholly or partly in shape-space, [16]-[18].

In what follows, we report on a multiple-parameterisation of the 8-cell model, originally suggested [15], and its ability to deal with layers of response in the human immune system. In Section 2, we summarise the basic model features and the growth and interaction rules of the system elements. Parameterisation is discussed in Section 3, and in Section 4 we report results obtained from the initial parameterisation and discuss performance sensitivity, w.r.t. [13], [19]. Our conclusions are presented in Section 5.

2 Eight Cell-Type Model of Immune Response

2.1 Cell Types and System Updating

The principal immune system features are represented here by eight cell-types. These differ from those proposed by Pandey [20], in that we ignore some less-highly active host cells in the cell-mediated response, but include humoral features, in particular memory cells, which influence future activation of CD4/CD8 cells, producing classes T_H1, T_H2 of helper T-cells. Specifically, cell types included are Macrophage, (M), T_H1, T_H2 cells, (T1,T2), cytotoxic T-cells, (CT), Memory T- and B- cells (MT, MB respectively), Antibody (AB) and Antigen or viral cells (V). Additional signals, such as controls on growth, cell elimination (apoptosis) and others are dealt with by introducing probabilistic parameters, (Section 3).

The human immune system is represented, in three-dimensions, by a simple cubic lattice of linear dimension L, with periodic boundary conditions. The lattice is randomly seeded with the 8 cell types, such that any number of cell-types, (≤ 8), can be present at the same site, but each site can contain at most one cell

of each type. Boolean variables are used to describe the states of the cells, where "true" represents the high concentration of a given cell-type and "false" the low concentration. All cells are stochastically updated, with a single update of the whole lattice equal to a Monte Carlo step, (MCS). This asynchronous updating is initiated by random selection of a site, repeated in excess of L^3 times for one MCS and for1000 MCS to represent progression of the disease. At each site of the lattice, cell states are simultaneously updated so that information on each interaction is stored in a temporary state. The final states after updating all cell types are thus independent of the sequence used.

2.2 Model Expressions

Cell Growth. For any site on the lattice, the state of a given cell-type at time $t+1$ will depend on the states of its co-occupying cell-types and the six nearest-neighbour sites. The growth of a single cell-type population is dependent on inter-site rules and may be summarized for the intermediate state S' of a given cell as:

$$\begin{aligned} &S'_{ic}(x,y,z,t+1) = S_{ic}(x,y,z,t).or.S_{ic}(x+1,y,z,t).or. \\ &S_{ic}(x,y+1,z,t).or.S_{ic}(x,y-1,z,t).or.S_{ic}(x,y,z+1,t).or. \\ &S_{ic}(x,y,z-1,t) \end{aligned} \tag{1}$$

where ic is the cell-type, and x, y and z the usual coordinates of the space. For any cell-type at a given site with a same type nearest-neighbour, having state "true" at t, the state of the given site will change to "true" at $t+1$. The equation, while similar to those of [13], [19], is not a guarantee of successful growth, since for each cell-type, this is assumed to be subject to a probability, $P_g(ic)$. Typically, the $P_g(ic)$ do not all have the same value, but depend on known characteristics of the cell. Similarly, in humoral response, immunological memory may be preserved by transformation of a percentage of B-cells into Memory B-cells (MB) with probability P_{B-MB}, thus maintaining a record of viral invasion characteristics, [21].

Inter-cell Interactions. The full set of interactions between the different cell-types at a given site can be written as:

$$\begin{aligned} &\text{(a)} \quad M(t+1) = M'.or.V'.and.[not[M'.and.V']] \\ &\text{(b)} \quad T1(t+1) = M'.and.[not[T1'.or.V']] \\ &\text{(c)} \quad T2(t+1) = [T2'.or.[V'.and.M']].and.[not[T2'.and.V']] \\ &\text{(d)} \quad CT(t+1) = V'.and.[M'.or.T'] \\ &\text{(e)} \quad B(t+1) = AB'.or.[[B'.or.MB'].and.V'] \\ &\text{(f)} \quad AB(t+1) = AB'.or.[[B'.or.MB'].and.V'] \\ &\text{(g)} \quad V(t+1) = [M.or.V'].and.[not[CT'.or.T1'].or.AB'] \end{aligned} \tag{2}$$

In brief, Equation (2)(a) indicates that an antigen (or virus) can induce a macrophage at a given site, where none is present and can destroy an already present macrophage (by infection). In Equations (2)(b) and (c), the joint presence of an antigen and macrophage can activate growth and differentiation of

T-cells and newly-generated T-cells differentiate as T_H1, T_H2 fractions. If no macrophage presents to the antigen, it can be treated as free and can kill both T_H1 and T_H2. From Equation (2)(d), it is clear that a cytotoxic T-cell grows when an antigen, macrophage and a T_H1 cell are all present at a given site. Further, the presence of a T_H2 T-cell and an antigen can induce the growth of a B-cell, Equation (2)(e). Antibody secretion is described by Equation (2)(f) since, when the antigen presents at a given site, both B and MB-cells can secrete antibody, with different probabilities and antibodies may thus spread to nearest-neighbour sites. The final equation describes the elimination of an antigen cell by a cytotoxic T-cell or antibody at the same site.

3 Key Model Parameters

For the inter-cell interactions, Equations (2), every contingency is governed by a probability, with a value close to 1 implying a high likelihood of occurrence. Equally, P_{ropT1} represents the proportion of newly generated T-cells, which differentiate into T_H1, while $(1 - P_{ropT1})$ produce T_H2 cells.

3.1 Death Rates

The unit simulation time-period is assumed to be equivalent to the smallest half-life of the eight entities in the model. If the death process is taken, for simplicity, to occur after cellular growth and inter-cellular interactions have taken place, then both new and original cells will be subject to the same death rate, with no account taken of the degradation of cell function in the model. (The memory-less assumption is not ideal, but is a useful first approximation). We define the probability of cell death in terms of the biological half-life (τ) as time required for half the number of a given cell type to be eliminated. For the period given by σ in the real immune system represented by one MCS, then

$$P_{death} = e^{\frac{-(ln2)\times o}{\tau}} \tag{3}$$

The half-life (in days) for each cell-type in the model here is given to be: M(10),T1(10), T2(10), CT(10), B(10), MB(400),AB(20), ([22] and references therein).

3.2 Cell Mobility

The mobility of both host and viral invader cells leads to increased interaction opportunities. Thus, as in [23], each cell is permitted, at the end of a MCS, to move to one of the six neighbouring sites with equal probability P_{mob}. The fundamental step of the mobility algorithm is the examination of neighbouring sites for presence of key cell-types, (Equation (2)). If conditions are met, the mobility step succeeds.

Table 1. Parameters in the 8-cell Model; (initial values as given in parentheses)

Parameter	Role	Parameter	Role
$P_g(ic)$	Cell growth rates, each cell-type ic (V; 0.2)	P_{B-AB}	(2)(f) AB secretion for B - cell (0.2)
P_{V-M}	(2)(a)-M stimulated (0.4)	P_{MB-AB}	(2)(f) - AB secretion for MB-cell(1.0)
$P_{V\times M}$	(2)(a) - M destroyed (0.4)	$P_{CT\times V}$	(2)(g) Virus killed by CT cell(1.0)
$P_{rop}T1$	(2)(b)(c) - as above (0.3)	$P_{AB\times V}$	(2)(g) destroyed by Antibody (1.0)
P_{M-TH}	(2)(b)(c) - as above (1.0)	P_{death}	as above
$P_{V\times TH}$	(2)(b)(c)-TH destroyed (1.0)	P_{mob}	as above
P_{T1-CT}	(2)(d) - CT stimulated (0.5)	P_{mut}	as above
P_{T2-B}	(2)(e) - B stimulated (1.0)	P_{B-MB}	Transfer rate B to MB cells (0.1)

3.3 Mutation

The most severe effect of HIV infection is the destruction of CD4 T-cells (T_H1 and T_H2). These cells stimulate production of cytotoxic T-cells and B-cells and, once disrupted, can no longer prevent the whole immune system from corruption. If the population of T_H1 cells is dominant, the system maintains *immunocompetency*, so that the density difference between the viral and T_H1 cells, $(N_{AG} - N_{T1})/L^3$ can be taken to reflect the current immune response status.

In the 8 cell-type mesoscopic model, we consider the mutation rate of HIV to be embedded in the inter-site interactions, where T_H1 cells are no longer able to recognise mutated viruses and activate cytotoxic T-cell growth. Effectively, this implies that the term V' in Equation (2)(d) is replaced by V'_{mut} at intra-site inter-cell stage. The possibility that mutation will occur is assigned a probability P_{mut}. A summary of the model parameters and initial values, together with their roles is given in Table 1. Conditions are otherwise as described in [15].

4 Results and Discussion

Principal results for immunological memory, and the roles of MT and MB cells, have been discussed previously, [15], and will not be revisited here. Instead we focus on the performance of the extended and basic models, the argument for enhanced mutation dependent on system fitness and the attempt to represent that fitness through choice of parameter values.

Comparison with Basic (4 cell-type) Models. The dependence on mutation rate of the growth pattern is illustrated in Fig. 1 for a range of P_{mut} values, focusing on viral and T_H1 density difference, for a low initial viral concentration, $N_{AG}/L^3 = 0.01$. In this instance, the full T-cell mediated response is induced

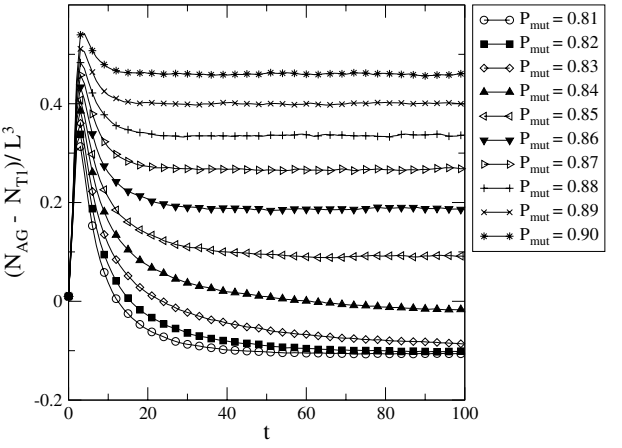


Fig. 1. 8-cell model : Density difference $(N_{AG} - N_{T1})/L^3$ vs t for range of P_{mut} values

with other parameters set at extreme values, (i.e. fully-activated growth, P_{death} = 0 and so on), for direct comparison with the basic models.An initial peak in the viral population is observed for all mutation rates considered, with a decrease over time as the immune defences respond. As P_{mut} increases to some critical value P_{crit}, however, the system moves from immuno-competence to sustained immuno-deficiency, as also found [13], [19] and, similarly, [24] for the basic models.

Averaging over ten to fifteen runs for every P_{mut} value considered, a plot of density difference vs P_{mut} indicates that a lower value of P_{crit} applies for the 8 cell-type model, compared to [19], with $P_{crit} = 0.842$ here, Fig. (2). (The subscript h refers to general Helper T-cells in the basic model and to T_H1 cells in our extended model). It seems clear that, in the latter case, the increased number and inter-dependencies of the cell-types lead to lower values of viral mutation being sufficient to overcome the host response and, further, that the sharpness of the transition is affected. Again, in line with findings for the basic models, [23], it is not possible to wholly decouple the effects of mobility and mutation in terms of the transition to immuno-deficiency. For extreme values of mobility, (0 and 1 for T_H1 and viral cells respectively), and the range of P_{mut} values considered, the density difference, $(N_{AG} - N_{T1})/L^3$, is much larger than for all cells immobile, with the virus dominating at an early stage. If host cells are assigned $P_{mob} > 0$, the implicit mobility of cell types via nearest-neighbour interactions is not altered, (Equation (2)), but can be thought of in terms of chemotaxis, or directed migration of cell-types. The maximum value of the difference in viral and T_H1 population densities is then strongly-dependent on the mutation rate, as found [23], with increased host mobility actually contributing to the spread of HIV infection,(not shown here). Restricting viral mobility alone, for all host cells mobile, leads to a reduction in the critical mutation value, but the phase transition is again noticeably blurred.

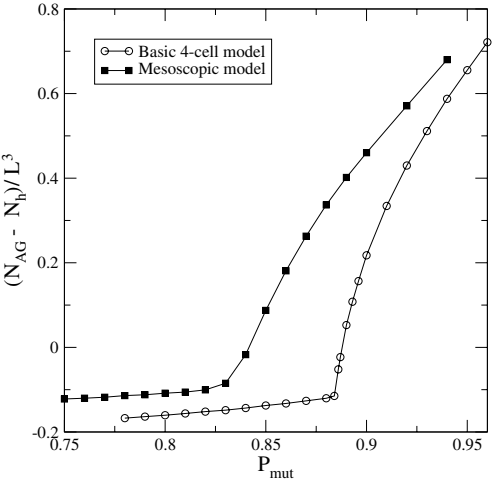


Fig. 2. Density difference between viral and helper cells for different P_{mut} values, P_{mob} = 0, illustrating shift in P_{crit} between 8-cell-type and basic models, [19]

4.1 Variable Mutation

The degree to which latency masks viral mutation activity is a key issue and in earlier work, [14], we considered potential for mutation enhanced by nearest-neighbour viral load. A modification of this is considered here, where the total viral mutation rate P_t consists of the initial mutation rate, plus an additional term at the chronic stage $P_\mu \propto N_{AG}/L^3$ (the viral density). Thus, the additional mutation component at time step t+1 depends on the global viral density at time t and P_t varies at every time step. For an initial mutation rate well below the critical value, ($P_{mut} = 0.81$), the three phases of HIV infection are readily reproduced with the virus peaking rapidly, followed by a decline to low level (chronic phase) for time $t \approx 50$, as the viral density reaches equilibrium. Application of the enhanced viral mutation rate leads to variable length of the chronic phase or latent period, with T_H1 cells maintaining dominance. In the final phase, as the system crosses into immuno-deficiency, the viral population level increases dramatically and T_H1 level drops equally rapidly as the host defense is overcome. This explosive viral growth is illustrated for enhanced mutation rates in Fig. 3, for $P_{mut} = 0.81$, where the latent period is about 600 MCS. Clearly this period is variable and will depend on factors such as growth rates, P_{mut} and the stochastic enhancement.

4.2 Moderation of Immune Response Due to Additional Cell-Types

For illustration here on the role of host cell-types, we focus on the Macrophage population dynamics. For the parameter values set for direct comparison with the basic model, the complementary dynamics of the host and virua cells are

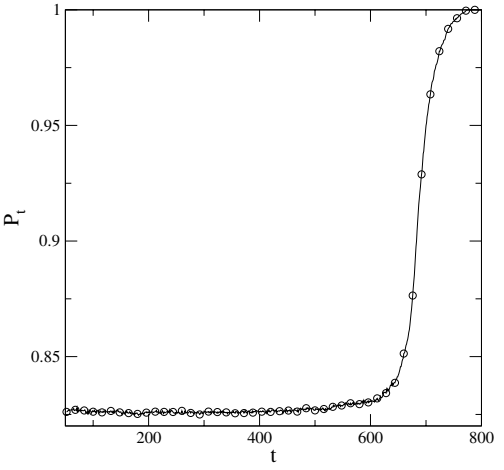


Fig. 3. Effect of variable mutation rate, with enhanced mutation $\propto$ global viral cell density [19]

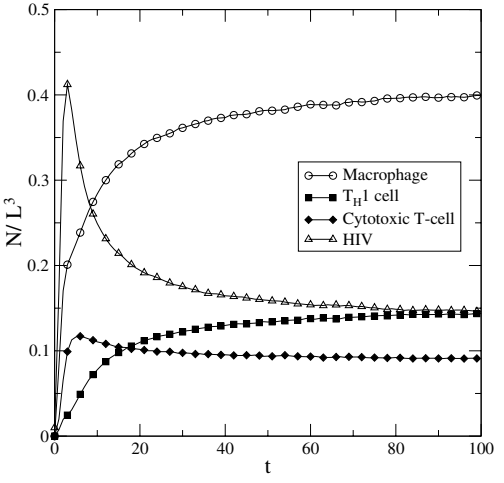


Fig. 4. Cell populations of immune cells and virus under primary HIV infection for P_{mut} just below the critical threshold. Other parameter settings as for basic model, [19].

illustrated at the primary stage in Fig. 4 with P_{mut} just below P_{crit}, $P_{mob} = 0$. As the viral threat declines (but does not disappear), so correspondingly does the cytotoxic cell population level, while $T_H1$cells match the viral level. The host system stays in control, with a high level of macrophages, reaching an equilibrium value, once the viral threat is past the peak.

The role of the macrophages is not, however, wholly beneficial. The invasion of antigens induces macrophage growth, in turn stimulating an increase in helper and killer cells, but the macrophages can themselves lead to increase in

Table 2. Parameter Values for auxiliary cell types

P_{V-M}	1.0	P_{M-Th}	1.0
P_{T1-CT}	1.0	P_{T2-B}	1.0
$P_{V\times TH}$	0.5	$P_{V\times M}$	0.5
P_{B-AB}	1.0	$P_{CT\times V}$	0.5
$P_{AB\times V}$	0.5	$Pg(ic)$ all	1.0

viral growth, through genetic materials available after infection. As an illustration, we can consider the case for macrophages under constant mutation rate, taking this initially to be some sub-critical value, $P_{mut} = 0.83$, (well within the immuno-competent range). In the intra-site interactions, Equation (1), a macrophage grows by the activation of an antigen at the same site with a probability, P_{V-M}. For the other parameters as given, Table 1, average densities for viral and Macrophage cells were simulated for a range of P_{V-M} values, where all densities reached an equilibrium value by around 1000 MCS. As P_{V-M} increases, the macrophage density rises until it reaches saturation. This saturated stage can only be observed for $P_{mut} < P_{crit}$, although as $P_{mut} \to P_{crit}$, the saturation level increases in order to meet the level of attack.

Once the critical threshold is reached, however, the macrophage population cannot evolve sufficiently rapidly to match explosive viral growth,which it is in part stimulating, i.e. it continues to increase indefinitely, but always lags behind. If the humoral arm of the immune response is also invoked, the host cells can further inhibit the overall effectiveness of the defense. For example, in Table 2, the parameter values given reflect a situation in which the parameters used to promote the growth of host support are fully activated, whilst the ones involved in suppression are partially activated.

It is also readily demonstrated, for example, that viral levels (for low (fixed) viral mutation rate and viral cells mobile) are highest when T_H1 production levels are lowest (i.e. $P_{ropT1} = 0.0$), and lowest for both arms of the immune response equally engaged, ($P_{ropT1} = 0.5$).

5 Conclusions

We demonstrate that the extended 8 cell-type model of immune response to HIV shares qualitative characteristics with similar models having fewer cell-types. In particular, critical mutation rates are comparable, with that for the extended model being slightly lower $P_{crit(4)} > P_{crit(8)} = 0.842$. Introducing multiple stochastic parameters indicates, however, that while basic models represent main features of the disease progression quite well, (permitting some aggregation of extended model components), sensitivity to system susceptibility (or fitness) is less well described. In particular, results indicate that host cells can inhibit the immune defense in a number of ways, not least in terms of actively promoting

viral growth, through the dual role of the macrophages. While a partial and qualitative picture only is presented here, the evidence suggests that de-stabilisation of the host system faciltates viral success, as much as the direct action of the virus in destroying a single key-cell type. This multiple attack programme, a well-known feature of HIV infection, is also strengthened if mutation rates are variable over time and related to fitness of the overall system. These findings are supported clinically by knowledge of variation in individual latency periods and are in agreement with recent work which suggests that the success of HIV is due, not only to T-cell depletion, but also to the joint failure of host cells to maintain homeostasis, [25].

References

1. Perelson A.S. ed. (1988) and articles with co-workers therein, Theoretical Immunology, Part 1 and 2. Addison-Wesley
2. Kaufman, M., Urbain J. and Thomas R. (1985). J. Theor. Biol. 114, 527
3. Seiden P.E. and Celada F.(1992). J. Theor. Biol. 158, 329.
4. de Boer R.J. , Segel L.A., Perelson A.S.(1992). J. Theor. Biol. 155, 295.
5. Fishman M.A. and Perelson A.S. (1994). J. Theor. Biol. 170, 25.
6. Bezzi M., Celada F., Ruffo S., Seiden P.E. (1997). Physica A 245, 145
7. Kohler B., Puzone R., Seiden P.E., Celada F. (2001). Vaccine 19, 862
8. Lagreca M.C., Almeida R.M.C., Zorzenon dos Santos R.M. (2001). Physica A 289, 191.
9. Pandey R.B. and Stauffer D. (1989). J. Phys. France 50, 1.
10. Kougias C.F. and Schulte J. (1990). J. Stat. Phys. 60, (1/2), 263.
11. Stauffer D., and Pandey R.B. (1992). Computers in Physics, 6, (4), 404
12. Mielke A. and Pandey R.B. (1998). Physica A 251, 430
13. Mannion R., Ruskin H.J. and Pandey R.B. (2000). Theory Biosci. 119, 145
14. Ruskin H.J., Pandey R.B. and Liu Y. (2002). Physica A 311, 213
15. Liu Y. and Ruskin H.J. (2002). Lecture Notes in Computer Science 2329, 127 (Springer-Verlag)
16. Perelson A.S. and Oster G.F. (1979). J. Theor. Biol. 81(4), 645.
17. Hershberg U., Louzon Y., Atlan H. and Solomon S. (2001). Physica A, 289, 178.
18. Burns J. and Ruskin H.J. (2003). Lecture Notes in Computer Science, 2660, 75 (Springer-Verlag)
19. Mannion R. Ruskin H.J. and Pandey R.B.(2002). Theory Biosci.121, 237
20. Pandey R.B. (1990). J. Phys. A.: Math. Gen. 23, 4321
21. Roitt I.M., Brostoff J., Male D.K. (2001). Immunology (6th Edn.) (Mosby Inc.)
22. Kleinstein S.H. and Seiden (2002). IMMSIM ++ Introduction and Users Guide
23. Pandey R.B., Mannion R. and Ruskin H.J. (2000). Physica A, 283, 447
24. Zorzenon dos Santos R.M. and Coutinho S.C. (2001). Phys. Rev. Lett. 87, 168
25. Bernaschi M. and Castiglione F. (2003). Immun. and Cell Biol. 80, 1.

Semantic Completeness in Sub-ontology Extraction Using Distributed Methods

Mehul Bhatt[1], Carlo Wouters[1], Andrew Flahive[1], Wenny Rahayu[1], and David Taniar[2]

[1] La Trobe University, Australia
{mbhatt,apflahiv,cewouter,wenny}@cs.latrobe.edu.au
[2] Monash University, Australia
David.Taniar@infotech.monash.edu.au

Abstract. The use of ontologies lies at the very heart of the newly emerging era of Semantic Web. They provide a shared conceptualization of some domain that may be communicated between people and application systems. A common problem with web ontologies is that they tend to grow large in scale and complexity as a result of ever increasing information requirements. The resulting ontologies are too large to be used in their entirety by one application. Our previous work, ***M****aterialized* ***O****ntology* ***V****iew* ***E****xtractor* (MOVE), has addressed this problem by proposing a distributed architecture for the extraction/optimization of a sub-ontology from a large scale base ontology. The extraction process consists of a number of independent optimization schemes that cover various aspects of the optimization process. In this paper, we extend MOVE with a Semantic Completeness Optimization Scheme (SCOS), which addresses the issue of the semantic correctness of the resulting sub-ontology. Moreover, we utilize distributed methods to implement SCOS in a cluster environment. Here, a distributed memory architecture serves two purposes: (a). Facilitates the utilization of a cluster environment typical in business organizations, which is in line with our envisaged application of the proposed system and (b). Enhances the performance of the computationally extensive extraction process when dealing with massively sized realistic ontologies.

Keywords: Parallel & Distributed Systems, Semantic Web, Ontologies, Sub-Ontology Extraction.

1 Introduction

The next generation of the internet is called the *semantic web*, and provides an environment that allows more intelligent knowledge management and data mining. The main focus is the increase in formal structures used on the internet. The taxonomies - with added functionality, such as inferencing - for these structures are called ontologies [1,2], and the success of the semantic web highly depends on the success of these ontologies. The reason ontologies are becoming popular

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 508–517, 2004.
© Springer-Verlag Berlin Heidelberg 2004

is largely due to what they promise: a shared and common understanding of a domain that can be communicated between people and applications.

A major problem is that as an ontology grows bigger, user applications only require particular aspects of the ontology as they do not benefit from the plethora of semantic information that may be present in the ontology. However, using the ontology means that all the drawbacks from this extra information are encountered; complexity and redundancy rise, while efficiency falls. This brings with it a clear need to create a sub-ontology [3,4]. For instance, if a business (application) only concerns itself with the efficiency of the workers, there is no need to access the detailed product catalog. Extracting just the part that is needed offers a smaller, more efficient, simpler solution/ontology.

A lot of research in similar areas has been done (e.g. in [5,6,7,8]). Previous research by the authors pioneered in the specialized area of ontology extraction [9,10]. An extraction methodology, consisting of a number of optimization schemes, was introduced to meet the extraction requirements, and guarantee a high quality resulting sub-ontology. However, this extraction process often proves to be computationally expensive, because ontologies in realistic settings turn out to be very large. For instance, the Unified Medical Language Systems (UMLS) base ontology has more than 800,000 concepts (nodes) and more than 9,000,000 relationships between those concepts).

This work was done as a part of a bigger project [11] involving materialized sub-ontology extraction using distributed methods. Distribution not only makes the process faster, but more importantly also facilitates our envisaged application of the extraction process. Often, business organizations have a cluster-like setup of inter-connected workstations as opposed to a single *shared-memory*, High-Performance Computing (HPC) facility. One reason for this is that a *'Beowulf Class Cluster'* setup is easily affordable than a centralized HPC facility. It is this setup that we aim to leverage upon by implementing a distributed memory architecture for the sub-ontology extraction process. In this paper, we look at the issue of semantic completeness of an extracted sub-ontology implemented in a distributed environment. As with all stages, even this stage will be referred to as an optimization scheme, hence the name Semantic Completeness Optimization Scheme (SCOS).

2 Previous Work: MOVE

Figure 1 shows a schematic of the sequential extraction process called Materialized Ontology View Extraction (MOVE) [11]. The process begins with the import of the ontology externally represented using XML. The actual extraction process/execution of optimization schemes is initiated by way of *requirements specification* by a user or another application. In the sub-sections that follow, each of the main components illustrated in Fig. 1 will be discussed briefly.

The ***'ontology import layer'*** (component 1) is responsible for handling various ontology representation standards that the extraction process is supposed to be compliant with. This is currently achieved in MOVE by transforming

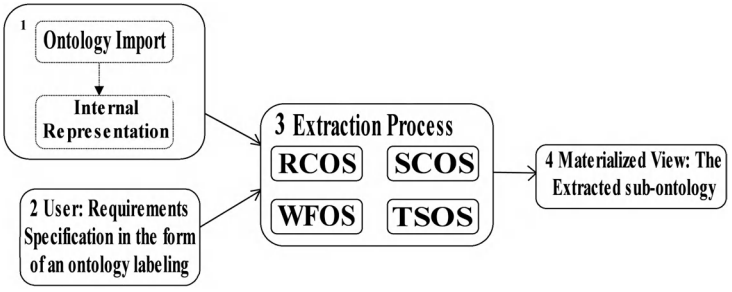


Fig. 1. The Sequential Extraction Process

the external representation of the ontology and its *meta-level* to a internal one that is specific to our implementation. It is necessary for user applications to use our import layer so as to be able to utilize the extraction algorithms. The *representation layer* maintains an object-oriented view of the ontology and its meta level. This facilitates easy extensibility as new ontology elements (new types) may easily be added in the ontology as well as its meta-level.

*'**Labeling**'* (component 2) of the base ontology facilitates user manipulation of the extraction process. The labeling may also be re-applied (i.e. modification of the user specified labeling) by the intermediate steps involved in the extraction process. This is the standard way different components of the extraction process (different extraction algorithms) may communicate with each other. Therefore, labeling is very crucial in the interaction between users & the extraction algorithms and algorithms amongst themselves. It allows a user to provide subjective information, pertaining to what must/must not be included in the target sub-ontology, on which the extraction process is based on. Moreover, an algorithm may work upon the labeling specified by the user, modify it in a certain way while preserving the semantics of the specification and pass it on to another algorithm within the extraction process. Currently, every ontological element may have a labeling of ***selected*** - must be present in the sub-ontology, ***deselected*** - must be excluded from the sub-ontology or *void* - the extraction algorithm is free to decide the respective elements inclusion/exclusion in the sub-ontology.

The *'**extraction process**'* (component 3) involves application of various optimization schemes that handle various issues pertaining to it such as ensuring consistency of initial requirements, well-formedness and deriving a sub-ontology that is highly qualitative in a sense that it is optimum and is the best solution to the users requirements. Note that the extraction process in not limited to optimization schemes currently being used in our framework. Also, it is possible that a particular scheme be completely left out of the it. Currently, the extraction process consists of Requirements Consistency Optimization Scheme (RCOS1-RCOS4), Semantic Completeness Optimization Scheme (SCOS1- SCOS3), Well Formedness Optimization Scheme (WFOS1-WFOS5) and Total Simplicity Optimization Scheme (TSOS1 - TSOS3). RCOS checks for the consistency of the user specified requirements for the target ontology and SCOS considers the complete-

ness of the concepts, i.e. if one concept is defined in terms of an another concept, the latter cannot be omitted from the sub-ontology without loss of semantic meaning of the former concept. It might be possible that the user requirements (labeling) is consistent. However, there might be statements that inevitably lead to a solution that is not a valid ontology. WFOS contains the proper rules to prevent this from happening. Applying TSOS to a existing solution will result in the smallest possible solution that is still a valid ontology.

The result of the extraction process is not just simply a extracted sub-ontology, but rather an extracted ***'materialized ontology view'*** (component 4) [9]. In the extraction process, no new information should be introduced (e.g. adding a new concept). However, it is possible that existing semantics are represented in a different way (i.e. a different view is established). Intuitively, the definition states that - starting from a base ontology - elements may be left out and/or combined, as long as the result is a valid ontology. In the process, no new elements should be introduced (unless the new element is a combination of a number of original elements, i.e. the *compression* of other elements). A *materialized* ontology view is required, as the resulting sub-ontology should be an independent ontology, i.e. *should* be a valid ontology even if the base ontology is taken away.

3 Semantic Completeness of a Sub-ontology

The idea of semantic completeness of an ontology can be interpreted in a number of ways. However, for the purposes to sub-ontology extraction, it amounts to the inclusion of the ***defining elements*** for the elements selected by the user by way of requirements specification. A defining element is a concept, relationship or attribute that is essential to the semantics of an another element of the ontology. A concept selected to be present in the sub-ontology would be semantically incomplete if its super-concept (the defining element in this case) is deselected at the same time. This could be further generalized into a situation where a set of elements are connected by a IS-A relationship unto any arbitrary depth. The scenario can only get more complex in the presence of more complex relationships such as multiple-inheritance, aggregation etc. The Semantic Completeness Optimization Scheme (SCOS) exists to guard against such inconsistencies. Below we present some notation consistent with [9], which is useful to define a common vocabulary pertaining to the ontological workload.

- $\delta_{\mathrm{B}}(\mathrm{b})$: Denotes a binary relationship between concepts
- $\delta_{\mathrm{C}}(\Pi_1(\mathrm{b}))$: First concept associated with $\delta_{\mathrm{B}}(\mathrm{b})$
- $\delta_{\mathrm{C}}(\Pi_2(\mathrm{b}))$: Second concept associated with $\delta_{\mathrm{B}}(\mathrm{b})$
- $\delta_{\mathrm{attr}}(\mathrm{t})$: Denotes a attribute-concept relationship
- $\delta_{\mathrm{C}}(\Pi_1(\mathrm{t}))$: A concept with associated attribute, ie: the concept in a $\delta_{\mathrm{attr}}(\mathrm{t})$
- $\delta_{\mathcal{A}}(\Pi_2(\mathrm{t}))$: An attribute with associated concept, ie:the attribute in a $\delta_{\mathrm{attr}}(\mathrm{t})$

Before we proceed with illustrating the distribution scheme, it is necessary that each of SCOS1-SCOS3 be defined, albeit informally for the purposes of

this paper. A formal introduction to SCOS (and the entire extraction process) along with a practical walk-through with intuitive examples can be found in [9]. SCOS1-SCOS3 are as follows: 1) **SCOS1**: If a concept is selected, all its super-concepts, and the inheritance relationships between the concepts and its super-concepts have to be selected. 2) **SCOS2**: If a concept is selected, all the aggregate part-of concepts of this concept, together with the aggregation relationship have to be selected as well. 3) **SCOS3**: If a concept is selected, all the attributes it possesses with a minimum cardinality other than zero and their attribute mappings should be selected as well.

4 SCOS: Proposed Distributed Implementation

SCOS1 and SCOS2 are conceptually similar with the difference that the former deals with a collection of inheritance relationships while the latter with a collection of aggregation relationships. This collection can be conceptualized as a forest of sparsely connected undirected graphs with the concepts representing the vertices and the relationships representing the edges of the graph. Such a conceptualization (& representation) using a graph-theoretic approach is optimal (& convenient) for purposes of distribution of SCOS. For example, consider checking the semantic for the completeness of a (potentially huge) set of concepts related by binary inheritance relationship, ie., SCOS1. If the set is to be partitioned & distributed to different processors so that SCOS1 may be run on each of the partitions in parallel, it is obviously desirable to allocate one connected component to each of the processors.

4.1 Problems with Graph Based Representation

A major problem with representing the data partitioning problem (for SCOS) in a graph-theoretic manner is that our underlying ontology representation is not graph theoretic. During ontology import, we construct a object oriented representation of the ontology as well as its meta-level. No structural information regarding the connectivity of the ontological elements is present. Since a ideal ontology would be massive in size and complex in structure, it would be optimal from a performance view-point that the graph based representation be constructed at the time of initial ontology import. Our object-oriented design represents a trade-off decision we took given the fact that other optimization schemes (such as RCOS) do not benefit from a graph based representation. So a graph based representation encompassing all different types of ontological elements would not be particularly useful.

4.2 Proposed Solution

Prior to data partitioning for SCOS, a graph based representation for elements specific to SCOS (binary inheritance & aggregation relationships) is constructed. We use the standard adjacency structure representation (comprised of adjacency

lists) to construct the ontoGraph. This involves additional work in the form of pre-processing of the SCOS workload to extract the concept set (ie. the vertices of the graph). This is necessary so as to construct the adjacency structure representation. Once the graph based representation is complete, we use a technique similar to a depth first search algorithm on the graph to get the set of connected components (called partitions hereafter) so as to schedule each of those for distribution to worker processors. Below, we discuss the three main steps, namely Ontology pre-processing, ontoGraph construction and Partition formation, involved in ontology pre-processing.

- **Ontology pre-processing:** The pre-processing phase basically involves constructing the vertex *set* so as to be used by the ontoGraph construction module. The input to this phase is the whole ontology. Processing begins by extracting the *list* of binary (inheritance & aggregation for SCOS1 & SCOS2 respectively) relationships from the ontology and inserting the elements related by each of the relationships in the list to a *set based container* thereby avoiding duplicates. Moreover, the unique vertices in the vertex set are keyed from 0 to N - 1, where N is the cardinality of the vertex set.
- **OntoGraph Construction:** As mentioned before, we represent the ontoGraph using the standard adjacency structure representation. The adjacency structure consists of a vector of lists of graph nodes. Each node in turn consists of other information such as an integral id of the ontology element is represents, a link to the element is represents etc. This ancillary information is necessary during the next phase, namely Partition Formation.
- **Partition Formation:** Partition formation in our case is equivalent to finding the different connected components in the ontoGraph. We currently use a technique similar to a depth first traversal (of the ontoGraph) to achieve this. The input to this phase is the ontoGraph whereas the result consists of a list of partitions. Here, one might get the impression that all the partitions need to be formed before any of them are assigned to worker processors. However in actuality, there is no reason to wait for the next partition to be generated before the current one is scheduled for distribution to a free processor as the partition sets are all going to be disjoint. As shall be illustrated later, we make use of asynchronous distribution primitives to assign the most recently generated partition to a free processor without waiting for the next one to be formed. This is advantageous as working out the semantic completeness (for the assigned partition) and formation of the next partition can proceed in parallel. The asynchronous nature of the primitives only adds to this optimality.

4.3 SCOS Distribution

For implementing the Requirements Consistency Optimization Scheme [9,11], we utilized a modified version of the classic task-farm model. SCOS essentially utilizes a similar distribution model with the exception that there is continued two-way interaction between the master & worker processors. Moreover, unlike

the RCOS distribution scheme, the worker processes do not need to post *updates* (requests for missing data) as they have all the data elements that would ever be needed to perform the most recently assigned task (ie: any of SCOS1-SCOS3). Also, data partitioning by the master & processing by the workers happens concurrently as the master dynamically creates partitions and assigns them to worker processes in a round-robin manner.

We use three asynchronous data distribution/result collection primitives namely ***gatherModifiedLabelingsFrom(...)***, ***recvPartition(...)*** and ***sendModifiedLabelings( )***. *recvPartition(...)* is used by the worker processes to receive the ontological workload that needs to be processed. Likewise, *gatherModifiedLabelingsFrom(...)* is used by the main processor to gather results from the worker processors, which they send using the *sendModifiedLabelings( )* primitive . As explained in section 2, this result takes the form of the modified labeling set. Note that it may be possible for the main processor to receive a 'semantic incompleteness' message and still get a modified labeling set. This is because following the rules for SCOS1-SCOS3, the worker processors attempt to make the extracted view as semantically complete as possible even if a 100% completeness is not possible.

Master processor execution consists of performing the necessary preprocessing of the ontology as explained in section 4.2. To re-iterate, it involves ontology initialization, extraction of the unique vertex & edge set, building the ontoGraph representation and perform the ontoGraph partitioning coupled with asynchronous distribution to the worker processors. Once the distribution is achieved, the only thing remains to be done for the master is collection and application of results to its solution set or the extracted view. As mentioned previously, irrespective of the results (ie: semantic completeness/incompleteness), the master always applies the labeling modifications worked out and serialized back

```
BEGIN Master
1. terminate = false;
2. resultsCounterSCOS1 = 0;
3. O = intialiseOntology();
4. V = preProcess(O);
5. G = buildOntoGraph(V);
6. P = partitionGraph(G);
7. partitionCount = |P| ;
8. while(terminate == false)
   {
      msg = getMessage();
      src = getSender();
      if(msg == SCOS1_SEMANTIC_COMPLETENESS)
        resultsCounterSCOS1++;
      else if(msg == SCOS1_SEMANTIC_INCOMPLETENESS)
        resultsCounterSCOS1++;

      modifiedLabelings = gatherModifiedLabelingsFrom(src);
      applyModifications(modifiedLabelings);

      if(resultsCounterSCOS1 == partitionCount)
        terminate = true;
   }
9. broadcastExitMessage(ALL_WORKERS);
END
```

```
BEGIN Worker
1. terminate = false;
2. while(terminate == false)
   {
      msg = getMessage();
      if(msg == SCOS_EXIT)
         teminate = true;
      else if(msg == SCOS1_PARTITION)
      {
         [δ_B(b)] = recvPartition(SCOS1_PARTITION)
         O = initLocalOntology([δ_B(b)]);
         Result = SCOS1_Single(O);
      }
      else if(msg == SCOS2_PARTITION)
      {
         [δ_attr(t)] = recvPartition(SCOS2_PARTITION)
         O = initLocalOntology([δ_attr(t)]);
         Result = SCOS2_Single(O);
      }

      sendMessage(Result);
      sendModifiedLabelings();
   }
END
```

Fig. 2. Master & Worker Processors

by the worker processes. Depending on the number of partitions scheduled to the worker processors, the master can figure out when results for each of them have been received and it is appropriate for the workers and itself to terminate.

Worker execution involves checking for a '*execution command*' from the main processor. Possible commands are to receive the workload pertaining to SCOS1-SCOS3 or actually execute SCOS1-SCOS3. Execution of any of the optimization schemes is followed by the sending back of results (modified labeling) and an indication of whether or not semantic completeness is possible for the most recently received partition. Note that the worker only terminates upon receiving an *'SCOS_EXIT'* message from the main processor.

4.4 Implementation

All implementation has been done using C++ on a Alphaserver SC supercomputer running Tru64 Unix 5.1. It has also been ported to a Linux Cluster environment with minimal modifications. Our distribution management component does not directly tackle issues pertaining to the cluster architecture, processor initialization etc. Instead, the *Messaging Passing Standard* [12], which encapsulates such architecture specific details and provides high level message-passing primitives suitable for distributed systems, has been utilized. As such, porting to other environments should not involve anything more than a recompilation on the target platform. Also, note that although homogeneous computing elements are being utilized currently, this is not a requirement for the implementation. Any distributed architecture is good enough as long as it supports the MPI message passing standard. It is up to the standard to handle the underlying architectural details pertaining to the cluster setup.

5 Evaluation

This section analyzes the results obtained from the distribution of the SCOS processing. Five different ontologies were used for testing the performance of SCOS. Initialization of the ontology is a constant time operation. As such the time taken to load each ontology, its meta-ontology and the associated user requirements (labeling) has been excluded. The results shown only include the time that the SCOS processing and distribution were active. The results have been split into two graphs to make it easier to show the similarities and differences between the five different sized ontologies. Fig. 3 - graph(a) shows that small ontologies (1-2000 concepts) with a greater number of processors, doesn't improve the overall time. From 3000 concepts and above, using a greater number of processors does speed up the time taken to complete the work. However, the larger ontologies in graph(b), seem to flatten off as the number of processors increase. This is where the added cost of communication for the extra processors out-weighs the amount of work left to be distributed. From these graphs then, we can suggest that for a certain ontology size, SCOS should employ a specific number of processors to do the work to obtain the most efficient use of resources.

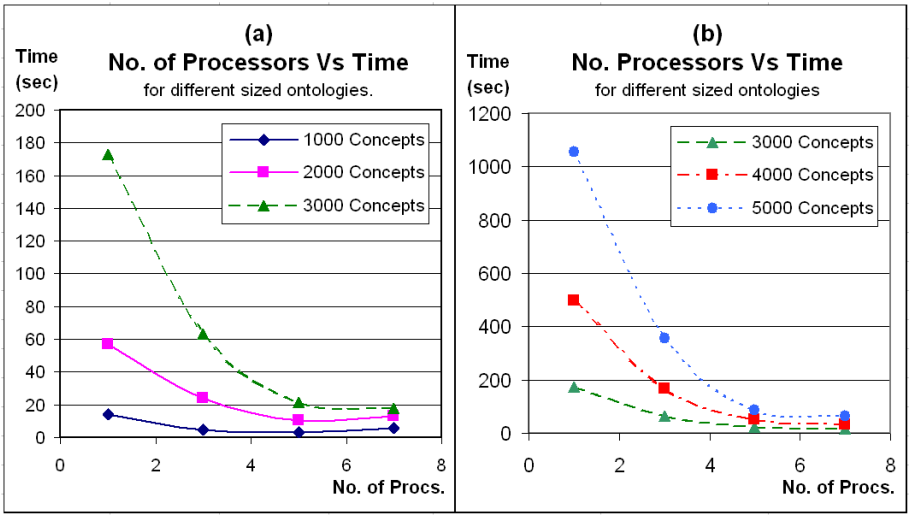


Fig. 3. SCOS Performance Results

6 Conclusion and Future Work

SCOS currently consists of three sub-schemes SCOS1-SCOS3, which handle various issues pertaining to semantic completeness. As mentioned previously, we have defined semantic completeness to be the inclusion/exclusion of certain *defining elements* for the ontological elements selected/deselected by the user or other application. Obviously, this notion of semantic completeness could be expanded and new rules could be specified by other researchers in the ontology domain. However, the distributed architecture that we have proposed and implemented is general enough to be used with other ontology based applications. The plugin based design of the optimization schemes as well as the distribution primitives facilitates seamless integration with other ontology applications.

The use of dynamic process management capability needs to be implemented into the system. Until the development of RCOS [11], the need for such capability within the framework did not arise as RCOS merely consists of partitioning the data to be processed based on the number of processors available. However, in the case of SCOS, the number of partitions generated is a property of the structure/connectivity of the OntoGraph. As such, a capability to allocate an optimal number of processors based on the number of data partitions will be necessary for optimized execution. Currently, we use a very coarse-grained distribution scheme. It is possible to utilize more sophisticated distribution schemes given the fact that there are no constraints as to the order of execution of functionally independent optimization schemes. Also, more fine-grained solution could consist of parallelization of individual sub-schemes within a particular optimization scheme. However, these enhancements would only be justified if optimal performance of the extraction process is a absolute

necessity. As previously mentioned, our research and the resulting distributed architecture is strongly driven by our envisaged application of the extraction process in a distributed business environment.

Acknowledgment. This work has been financially supported by Victorian Partnership For Advanced Computing (VPAC) Expertise Grant Round 5, No:EPPNLA090.2003. All implementation was done using VPAC's supercomputing facilities.

References

1. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. In: Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, Deventer (1993)
2. Guarino, N., Carrara, M., Giaretta, P.: An ontology of meta-level categories. In: KR'94: Principles of KRR. Morgan Kaufmann (1994)
3. Wouters, C., Dillon, T., et. al: A practical walkthrough of the ontology derivation rules. In: Proceedings of DEXA 2002, Aix-en-Provence (2002)
4. Spyns, P., Meersman, R., Mustafa, J.: Data modelling versus ontology engineering. SIGMOD (2002)
5. Guarino, N., Welty, C.: Evaluating ontological decisions with ontoclean. Communications of the ACM **45** (2002)
6. Klein, M., Fensel, D., Kiryakov, A., Ognyanov, D.: Ontology versioning and change detection on the web. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Volume 2473 of LNAI., Spain, Springer Verlag (2002) 197–212
7. McGuinness, D.L., et. al.: An environment for merging and testing large ontologies. In: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning, San Francisco, Morgan Kaufmann (2000)
8. Noy, N.F., Klein, M.: Ontology evolution: Not the same as schema evolution. Technical Report SMI-2002-0926, Stanford Medical Informatics (2002)
9. Wouters, C., Dillon, T., Rahayu, W., et. al: A practical walkthrough of the ontology derivation rules. DEXA2002 (2002) 259–268
10. Wouters, C., Dillon, T., Rahayu, W., et. al: A practical approach to the derivation of materialized ontology view. In: Web Information Systems. Idea Group Publishing (2004)
11. Bhatt, M., Flahive, A., Wouters, C., Rahayu, W., Taniar, D., Dillon, T.: A Distributed Approach to Sub-Ontology Extraction. In: Proceedings of the Eighteenth International Conference on Advanced Information Networking and Applications (AINA'04), Fukuoka, Japan (2004)
12. William Gropp, Ewing Lusk, A.S.: Using MPI: Portable Parallel Programming with the Message-Passing Interface. Second edn. MIT Press (1999)

Distributed Mutual Exclusion Algorithms on a Ring of Clusters

Kayhan Erciyes

California State University San Marcos,
Computer Science Dept., 333 S.Twin Oaks Valley Rd.,
San Marcos CA 92096, U.S.A.
kerciyes@csusm.edu

Abstract. We propose an architecture that consists of a ring of clusters for distributed mutual exclusion algorithms. Each node on the ring represents a cluster of nodes and implements various distributed mutual exclusion algorithms on behalf of any member in the cluster it represents. We show the implementation of Ricart-Agrawala and a Token-based algorithm on this architecture. The message complexities for both algorithms are reduced substantially with this architecture as well as obtaining better response times due to parallel processing in the clusters ...

1 Introduction

Mutual exclusion in distributed systems is a fundamental property required to synchronize access to shared resources in order to maintain their consistency and integrity. Comprehensive surveys about mutual exclusion are given in [8] [1]. For a system with N processes, competetive algorithms have message complexities between $log(N)$ and $3(N-1)$ messages per access to a critical section (CS). The distributed mutual exclusion algorithms may be broadly classified as permission based or token based. In the first case, a node would enter a critical section after receiving permission from all of the nodes in its set for the critical section. In the second case, the posession of a system-wide unique token would provide the right to enter a CS. For token based algorithms, a logical ring of processes may be constructed and a token is passed around the ring. The token holder gets the permission to access the critical section. These types of algorithms are considered fair, and have bounded waiting. The token based approach is highly susceptable to the loss of a token as this would result in a deadlock. Susuki-Kasami's algorithm [10] (N messages), Singhal's heuristic algorithm [9] (N/2, N messages) and Raymond's tree based algorithm [5] (log(N) messages) are examples of token based mutual exclusion algorithms. Examples of nontoken-based distributed mutual exclusion algorithms are Lamport's algorithm [3] (3(N-1) messages), Ricart-Agrawala (RA) algorithm (2(N-1) messages) [6] and Maekawa's algorithm ($\sqrt{N}$ messages) [4].

The requirements for any distributed mutual exclusion algorithm are 1. At most one process should be executing in the critical section (safety), 2. request to

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 518–527, 2004.
© Springer-Verlag Berlin Heidelberg 2004

enter or exit the critical section will eventually succeed (liveness) and 3. If one request is issued before another, then the requests will be served in the same order (fairness). Lamport's algorithm [3] and RA algorithm [6] are considered as one of the only fair distributed mutual exclusion algorithms in literature. Lamport's algorithm requires 3(N-1) messages and RA algorithm optimizes Lamport's algorithm and requires 2(N-1) messages per critical section access.

In this study, we propose an architecture where coordinators for clusters of nodes are placed on a ring. These coordinators perform the required critical section entry and exit procedures for the nodes they represent. This model is semi-distributed as we have central components. However, these components are homogenous and communicate with each other asynchronously. Using this architecture, we show that RA and Token-based algorithms may achieve an order of magnitude reduction in the number of messages required to execute a critical section at the expense of increased response times and synchronization delays. The rest of the paper is organized as follows. Section 2 is a review on the background of performance metrics for fundamental mutual exclusion algorithms. The RA algorithm on the proposed model, *RA_Ring* is described in Section 3 with the analysis of the achieved performance metrics. The second algorithm implemented on the model uses Token Passing and is called *Ring_TP* as described in Section 4 with performance considerations. Finally, discussions and conclusions are outlined in Section 5.

2 Background

2.1 Performance Metrics

Performance of a distributed mutual exclusion algorithm depends on whether the system is *lightly* or *heavily* loaded. If no other process is in the critical section when a process makes a request to enter it, the system is lightly loaded. Otherwise, when there is a high demand for the critical section which results in queueing up of the requests, the system is said to be heavily loaded. The important metrics to evaluate the performance of a mutual exclusion algorithm are the number of messages per request, response time and the sysnchronization delay as described below :

- *Number of Messages per Request* (M) : The total number of messages required to enter a critical section is an important and useful parameter to determine the required network bandwidth for that particular algorithm. M can be specified for high load or light load in the system as M_{heavy} and M_{light}.
- *Response Time* (R) : The Response Time R is measured as the interval between the request of a node to enter critical section and the time it finishes executing the critical section. When the system is lightly loaded, two message transfer times and the executon time of the critical section suffices reulting in $R_{light} = 2T + E$ units. Under heavy load conditions, assuming at least one message is needed to transfer the access right from one node to another, $R_{heavy} = w(T + E)$ where w is the number of waiting requests.

– *Synchronization Delay* (S) : The synchronization delay S is the time required for a node to enter a critical section after another node finishes executing it. The minimum value of S is one message transfer time T since one message suffices to transfer the access rights to another node.

The lower bounds for M, R and S are shown in Table 1.

Table 1. Lower Bounds for Performance Metrics [7]

M_{light}	M_{heavy}	R_{light}	R_{heavy}	S
3	3	$2T + E$	$w(T + E)$	T

2.2 Ricart-Agrawala Algorithm

The Ricart-Agrawala (RA) Algorithm represents a class of decentralized, permission based mutual exclusion algorithms. In RA, when a node wants to enter a critical section, it sends a timestamped broadcast *Request* message to all of its peers in that critical section request set. When a node receives a *Request* message, it returns a *Reply* message if it is not in the critical section or requesting it. If the receiving node is in the critical section, it does not reply and queues the request. However, if the receiver has alraedy made a request, it compares the timestamp of its request with the incoming one and replies the sender if the incoming request has a lower timestamp. Otherwise, it queues the request and enters the criticals ection. When a node leaves its critical section, it sends a reply to all the deferred requests on its queue which means the process with the next earliest request will now receive its last reply message and enter the ctitical section. The total number of messages per critical section is 2(N-1) as (N-1) requests and (N-1) replies are needed. One of the problems with this algorithm is that if a process crashes, it fails to reply which is interpreted as a denial of permission to enter the critical section, so all other processes that want to enter are blocked. Also, the system should provide some method of clock synchronization between processes. The performance metrics for the *RA* Algorithm are shown in Table 2. When a node finishes execution of a critical section, one message is adequate for a waiting node to enter, resulting in $S = T$.

Table 2. Performance Metrics for Ricart-Agrawala Algorithm

M_{light}	M_{heavy}	R_{light}	R_{heavy}	S
$2(N - 1)$	$2(N - 1)$	$2T + E$	$w(T + E)$	T

2.3 Token-Based Algorithms

The general Token Passing (TP) Algorithm for mutual exclusion is characterized by the existence of a single token where the posession of it denotes permission to

enter a critical section. The token circulation can be performed in a logical ring structure or by broadcasting [10]. In a ring based TP Algorithm, any process that requires its critical section will block the token and issue it when it finishes executing. Fairness is ensured in this algorithm as each process waits at most $N-1$ entries to enter the critical section. There is no starvation since passing is in strict order. The main difficulties with TP Algorithm are as follows. There would be the idle case of no processes entering CS would incur overhead of constantly passing the token. There could be lost tokens which would require diagnosis and creating a new token by a central node or distributed control is needed and to prevent duplicate tokens, central coordinator should ensure generation of only one token. Crashes should also be dealt with as these would require detection of the dead destinations in the form of acknowledgements. One important design issue with TP Algorithm is the determination of the holding time for unneeded token. If this time is too short, there will be high overhead. However, keeping this time too long would result in high CS latency. The performance metrics for a general Token-Based Algorithm is shown in Table 3. We assume a general case here where $N-1$ messages to solicit for the token and 1 reply message from the holder are needed.

Table 3. Performance Metrics for General Token-based Algorithms

M_{light}	M_{heavy}	R_{light}	R_{heavy}	S
N	N	$2T+E$	$w(T+E)$	T

3 Ricart-Agrawala Algorithm on the Ring

We propose the architecture shown in Fig. 1 where nodes form clusters and each cluster is represented by a *coordinator* in the ring. Coordinators are the interface points for the nodes to the ring and election of a new coordinator is provided as in [2] if it crashes. The relation between the cluster coordinator and an ordinary node is similar to a central coordinator based mutual exclusion algorithm. The types of messages exchanged are *Request*, *Reply* and *Release* where a node first requests a critical section and upon the reply from the coordinator, it enters its critical section and then releases the critical section. However, the coordinator in this case has to provide more sophisticated functionality as it has to communicate with the other coordinators.

The finite state diagram of the coordinator is depicted in Fig. 2. When a node makes a request for a critical section ($Node_Req$), the coordinator sends a critical section request ($Coord_Req$) to the ring and sets its state to WAITRP. If there are any other requests from its cluster in this state, it sends a request message ($Coord_Req$) to the ring for each one and store these pending requests in its cluster. When it receives external requests ($Coord_Req$) in this state, it performs the operation of a normal RA node by checking the timestamps of the incoming requests by the pending requests in its cluster and sends a reply

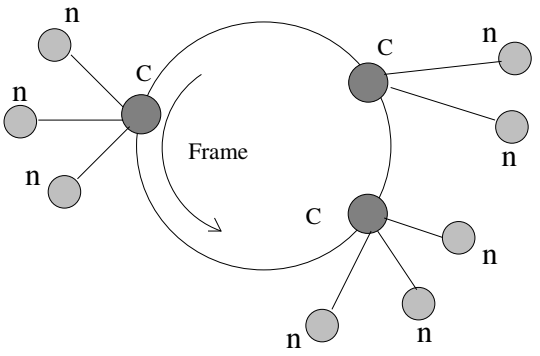


Fig. 1. The Architecture, n: Node, C: Coordinator

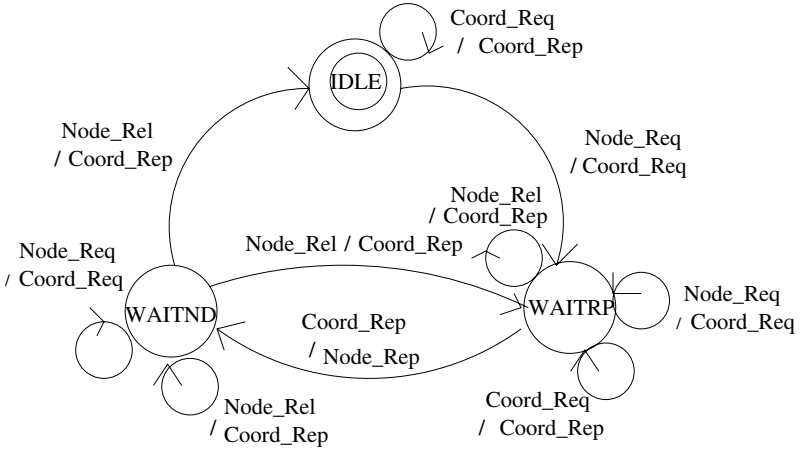


Fig. 2. The Coordinator for the Ring_RA Algorithm

(*Coord_Rep*) only if all of the pending requests have greater timestamps than the incoming request. It goes back to the IDLE state only if there are not any other pending request in its cluster. For any pending request, it goes back to the WAITRP state to wait for coordinator replies.

Fig. 3 shows an example scenario for the Ring_RA Algorithm. The following describes the events that occur :

1. Nodes n_{13}, n_{33} and n_{12} in clusters 1 and 3 make critical section requests with messages $req_1(13, 1)$, $req_2(33, 2)$ and $req_3(12, 3)$ respectively where the first parameter is the identity and the second is the timestamp of the request.
2. The coordinator for Cluster 1, C_1, forms two request messages R_{12} and R_{13} for each request and sends these to the next coordinator on the ring, C_2
3. C_2 passes these messages immediately to its successor C_3 as it has no pending requests in its cluster.

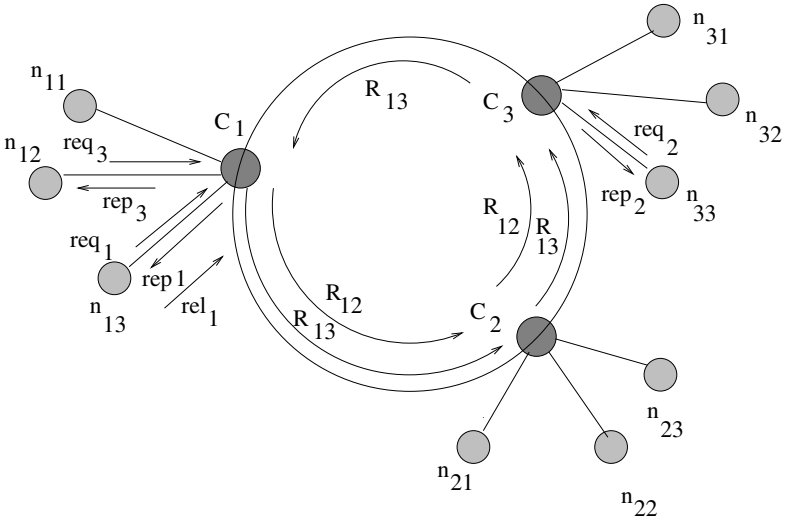


Fig. 3. Operation of the Ring_RA Algorithm

4. C_3 however has a pending request from n_{33} and checks the timestamps of the incoming requests with the timestamp of the request in its cluster. C_3 replies to R_{13} as it has a lower timestamp than n_{33} request by simply filling the acknowledgement field in the message but defers the reply to R_{12} as it has a greater timestamp.
5. C_1 receives the reply message for R_{13} and therefore sends a *Reply* message to n_{13} which enters its critical section.
6. C_3 has a similar request (R_{33}) in the ring which is blocked by C_1 because n_{13} has a lower timestamp.
7. When n_{13} finishes execution of its critical section, it sends a *Release* (rel_1) message to C_1 which in turn passes the blocked *Reply* message for n_{33} to C_2
8. C_3 now has the *Reply* message and n_{31} executes similar to Step 5.
9. C_3 now releases the *Reply* message for n_{12} to execute its critical section.

The order of execution in this example is $n_{13} \rightarrow n_{33} \rightarrow n_{23}$ in the order of the timestamps of the requests.

Theorem 1. *The total number of messages per critical section using the Ring_RA Algorithm is $k + 3$ where k is the number of coordinators (clusters).*

Proof. An ordinary node in a cluster requires three messages (*Request, Reply and Release*) per critical section to communicate with the coordinator. The full circulation of the coordinator request (*Coord_Req*) requires k messages resulting in $k + 3$ messages in total.

Corollary 1. *The Synchronization Delay (S) in the Ring_RA Algorithm varies from $2T$ to $(k + 1)T$ where k is the number of clusters.*

Proof. When the waiting and the executing nodes are in the same cluster, the required messages between the node leaving its critical section and the node entering are the *Release* from the leaving node and *Reply* from the coordinator resulting in two message times ($2T$) for S_{min}. However, if the nodes are in different clusters, the *Release* message has to reach the local coordinator, circulate the ring through $k-1$ nodes to reach the originating cluster coordinator in the worst case and a *Reply* message from the coordinator is sent to the waiting nodes resulting in S_{max}=$(k+1)T$.

Corollary 2. *In the Ring_RA Algorithm, the response times are* R_{light}=$(k+3)T+E$ *and* R_{heavy} *varies from* $w(2T+E)$ *to* $w((k+1)T+E)$ *where* k *is the number of clusters and* w *is the number of pending requests.*

Proof. According to Theorem 3, the total number of messages required to enter a critical section is $k+3$. If there are no other requests, the response time for a node will be R_{light}=$(k+3)T+E$ including the execution time (E) of the critical section. If there are w pending requests at the time of the request, the minimum value R_{heavy_min} is $w(2T+E)$. In the case of S_{max} described in Corollary 1, R_{heavy_max} becomes $w((k+1)T+E)$ since in general R_{heavy}=$w(S+E)$.

Since the sending and receiving ends of the algorithm are the same as of RA algorithm, the safety, prevention of starvation and the fairness attributes are the same. The performance metrics for the Ring_RA Algorithm are given in Table 4.

Table 4. Performance of the Ring_RA Algorithm

M_{light}	M_{heavy}	R_{light}	R_{heavy_min}	R_{heavy_max}	S_{min}	S_{max}
$k+1$	$k+1$	$(k+3)T+E$	$w(2T+E)$	$w((k+1)T+E)$	$2T$	$(k+1)T$

4 The Token Passing Mutual Exclusion Algorithm on the Ring

We propose a practical implementation of TP Algorithm to be executed using the architecture described in Section 3. In this algorithm, the token is circulated in the ring only. The coordinator for each cluster determines whether to consume the token or not, based on its state. The FSM diagram of the TP Coordinator is depicted in Fig.4. When a node wants to enter a critical section, it sends a request message to the coordinator which records this event and changes its state to WAITTK to wait for the token. Once it has the token from the ring, it sends this token to the node by *Coord_Tok* to grant the request of the node and changes its state to WAITND to wait for the node to finish its critical section. A finished node will then send the token *Node_Tok* to the coordinator which will then release the token for circulation in the ring or send it to another waiting node in its cluster as shown in Fig. 4.

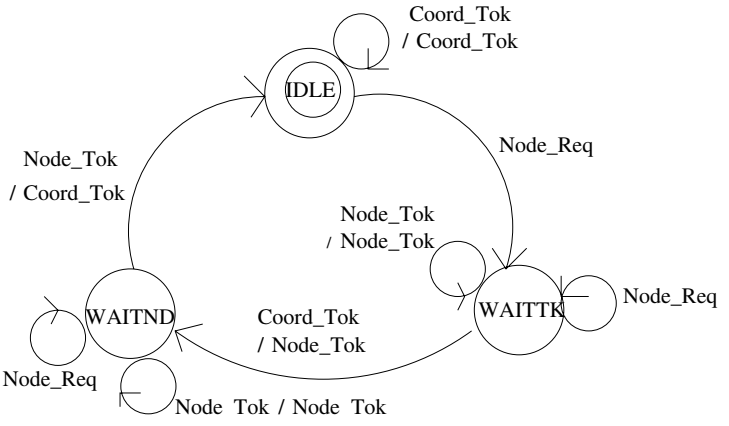


Fig. 4. The Coordinator for the Ring_Token Algorithm

Theorem 2. *The Ring_TP Algorithm has a messages complexity of* $O(k+1)$ *per critical section.*

Proof. The proof is similar to the proof of Theorem 1 except that $k+3$ is an upper bound on te number of messages depending on the current location of the token when a request is made.

The Synchronization Delay (S) and the Response Time values are similar to the Ring_RA Algorithm as shown in Table 5.

Table 5. Performance of the Ring_TP Algorithm

M_{light}	M_{heavy}	R_{light}	$R_{heavy-min}$	$R_{heavy-max}$	S_{min}	S_{max}
$O(k+3)$	$O(k+3)$	$(k+3)T+E$	$w(2T+E)$	$w((k+1)T+E)$	$2T$	$(k+1)T$

Table 6. Comparison of the Ring Mutual Exclusion Algorithms with others

	Regular	Ring Algorithms	Ring (k=m)	Gain(large k,m)
Ricart-Agrawala Alg.	$2(N-1)$	$k+3$	$O(\sqrt{N})$	$O(2m)$
Token Passing Alg.	N	$O(k+3)$	$O(\sqrt{N})$	$O(m)$

5 Discussions and Conclusions

We proposed an architecture to implement distributed mutual exclusion algorithms. We showed that this architecture provides improvements over message complexities of Ricart and Agrawala and Token-based algorithms and also the time required to execute a critical section.

A comparison of the two algorithms with their regular counterparts in terms of their message complexities is shown in Table 6. Here we see that it is possible to obtain an order of magnitude of improvement over the classical RA and the Token-Based algorithms using our model at the expense of large response times and increased synchronization delays. For large k values, the gains with respect to normal algorithms are shown in the last column of the table as $O(2m)$ and $O(m)$. This may be interpreted as the more nodes a cluster has, the less number of messages required to enter a critical section with respect to the regular algorithms. However, large k and m values would result in a coordinator becoming a bottleneck as in the central coordinator case and the response time would be large. The coordinators have an important role and they may fail. New coordinators may be elected and any failed node member can be excluded from the cluster which is an improvement over both classical algorithms as they do not provide recovery for a crashed node in general. The recovery procedures can be implemented using algorithms as in [2] [11] which is not discussed here. One other advantage of the proposed model is that the pre-processing of the requests of the nodes by the coordinators are performed independently resulting in improved performance.

Our work is ongoing and currently we are investigating the implementation of Susuki and Kasami algorithm [10] and Maekawa's algorithm [5] on this architecture. We are also looking into implementing k-way distributed mutual exclusion where there may be k nodes executing a critical section at one time. One other direction of study we are pursuing is the implementation of this model in mobile ad hoc networks. The mobile network can be represented as a graph which can be partitioned into a number of clusters periodically or upon change of configuration, using suitable heuristics. Once the partitioning is completed, ecah cluster can be represented by a coordinator and the model proposed will be valid to provide distributed mutual exclusion in mobile networks.

References

1. Chang.,Y.I.: A Simulation Study on Distributed Mutual Exclusion. Journal of Parallel and Distributed Computing, Vol. 33(2). (1996) 107–121
2. Erciyes, K.: Implementation of A Scalable Ring Protocol for Fault Tolerance in Distributed Real-Time Systems. Proc. of Computer Networks Symposium BAS 2001. (2001) 188–197
3. Lamport, L.: Time, Clocks and the Ordering of Events in a Distributed System. CACM, Vol. 21. (1978) 558–565
4. Maekawa, M.: A sqrt(n) Algorithm for Mutual exclusion in Decentralized Systems. ACM Transactions on Computer Systems, Vol. 3(2). (1985) 145–159
5. Raymond, K.: A tree-based Algorithm for Distributed Mutual Exclusion. ACM Trans. Comput. Systems, Vol. 7(1). (1989) 61–77
6. Ricart, G., Agrawala, A.: An Optimal Algorithm for Mutual Exclusion in Computer Networks. CACM, Vol. 24(1). (1981) 9–17
7. Shu, Wu, An Efficient Distributed Token-based Mutual Exclusion Algorithm with a Central Coordinator, Journal of Parallel and Distributed Processing. Vol. 62-10. (2002) 1602–1613

8. Singhal, M.: A Taxanomy of Distributed Mutual Exclusion. Journal of Parallel and Distributed Computing, Vol. 18(1). (1993) 94–101
9. Singhal, M.: A Dynamic Information Structure Mutual Exclusion Algorithm for Distributed System. IEEE Trans. Parallel and Distributed Systems, Vol. 3(1). (1993) 94–101
10. Susuki, I., Kasami, T.: A Distributed Mutual Exclusion Algorithm. ACM Trans. Computer Systems, Vol. 3(4). (1985) 344–349
11. Tunali, T, Erciyes, K., Soysert, Z.: A Hierarchical Fault-Tolerant Ring Protocol For A Distributed Real-Time System. Special issue of Parallel and Distributed Computing Practices on Parallel and Distributed Real-Time Systems, Vol. 2(1). (2001) 47–62

A Cluster Based Hierarchical Routing Protocol for Mobile Networks

Kayhan Erciyes and Geoffrey Marshall

California State University San Marcos,
Computer Science Dept., 333 S.Twin Oaks Valley Rd.,
San Marcos, CA 92096, U.S.A.
{kerciyes,marsh021}@csusm.edu

Abstract. We propose a hierarchical, cluster based semi-distributed routing protocol for a mobile ad hoc network. The network graph is partitioned into clusters using heuristics and the shortest routes are first calculated locally in each cluster in the first phase. The network is then simplified to consist only of the nodes that have connections to other clusters called the neighbor nodes and the shortest routes are calculated for this simple network in the second. A complete route between two nodes of different clusters is formed by the union of intra-cluster and inter-cluster routes. We show that this method has better performance with respect to the other methods of calculation of all-pairs shortest paths in a mobile network . . .

1 Introduction

Two general ways for building routing tables in an arbitrary computer network are the central and distributed approaches. In the central approach, connectivity information from all nodes are gathered in a central coordinator which performs some routing algorithm and distributes the routing tables to individual nodes. Dijkstra's *All-pairs Shortest Paths* (APSP) algorithm [2] uses the greedy approach and finds all routes in $O(n^3)$ time. Floyd and Warshal algorithm [4] uses dynamic programming and finds the routes similarly in $O(n^3)$ time. Distributed routing algorithms assume that there is no central component and global information available to the nodes of the network. Each node makes use of its local information and information it receives from its neighbors to find the shortest routes. Mobile ad hoc networks do not have central administration or fixed infrastructure and consist of mobile wireless nodes that have temporary interconnections to communicate over packet radios. The rapidly changing topology of a mobile network requires that routes should be calculated much more frequently than the wired networks. Distributed, adaptive and self-stabilizing algorithms may be used to perform routing in mobile networks. *Link reversal routing* algorithms are one class of such algorithms where a node reverses its incident links when it loses routes to the destination. Performance analysis of link reversal algorithms are given in [1] and TORA [9] is an example system that uses link reversal routing. An important routing approach in mobile networks is

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 528–537, 2004.
© Springer-Verlag Berlin Heidelberg 2004

clustering, that is, partitioning of the network into smaller subnetworks to limit the amount of routing information stored at individual nodes. In [8], a mobile network is partitioned into clusters of a two level graph. In the *zone routing* proposed in [5] where a zone functions similar to a cluster, the requested routes are first searched within the local zone. For inter-zone routes, the search is carried by multicast messages to the boundary nodes within the zones. In *k-way clustering*, the mobile network is divided into non-overlapping clusters where two nodes of a cluster are at most k hops away from each other. A k-way clustering method is proposed in [3] where the spanning tree of the network is constructed in the first phase and this tree is partitioned into subtrees with bounded diameters in the second phase.

In this study, we propose a hierarchical, semi-distributed, two-level dynamic distributed routing protocol for a mobile network. The protocol is not fully distributed due to the existence of some privileged nodes in the network. The distributed routing architecture consists of hierarchical clusters of routing nodes and each cluster has a controller which is called the *representative*. At the highest level, one of the representatives called the *coordinator* has the complete connectivity information of all the nodes in the network. Everytime there is an addition or deletion of a node to a cluster, the coordinator is informed to update its view. Upon such changes of configuration or periodically gathering of the changes, the coordinator starts a new configuration process by partitioning the network graph into new clusters. The nodes in the cluster including the nodes that have connections to other clusters are called the *neighbor nodes*. The coordinator chooses one of the neighbor nodes in each cluster as the cluster representative and sends the cluster connectivity information and neighbor connectivity information to the representative of such a group. Each representative then distributes the local connectivity information to all of the nodes in its group which concludes the first phase of the protocol. In the second phase, each node performs APSP routing within its cluster. This phase is concluded by calculating the distances between all pairs of nodes in the cluster including the neighbor nodes. In the third phase, only the neighbor nodes calculate all-pairs shortest path routes for the *neighborhood graph* which represents the simplified inter-cluster connectivity of the original network. Any route is then formed by the union of the route from the source node to its nearest neighbor, the shortest route between the source neighbor and the destination neighbor and the shortset route between the destination neighbor and the destination node.

The rest of the paper is organized as follows. The partitioning of the network is described in Section 2, the distributed route management is explained in Section 3 and an example network is detailed in Section 4. The performance analysis of the overal system is given in Section 5. Finally, the implementation carried so far is presented in Section 5 and future directions are outlined in the Conclusions section.

2 The Partitioning of the Network

The aim of any partitioning algorithm is to provide subgraphs such that the number of vertices in each partition is averaged and the number of edges cut between the partitions is minimum with a total minimum cost. Various partitioning algorithms for graphs for task scheduling and related problems exist. An arbitrary network can be constructed as an undirected connected graph $G = (V, E, w)$ where V is the set of routing nodes, E is the set of edges giving the cost of communication between the routing nodes and w: $E \rightarrow \Re$ is the set of weights associated with edges. *Multilevel partitioning* is performed by coarsening, partitioning and uncoarsening phases [6]. During the coarsening phase, a set of smaller graphs are obtained from the initial graph $G_i = (V_i, E_i, w_i)$ such that $|V_i| > |V_{i+1}|$. When graph G_{i+1} is to be constructed from graph G_i, a maximal matching $M_i \subset E_i$ is found and vertices that are incident on both edge of this matching are collapsed. The collapsing is performed as follows. If $u, v \in V_i$ are collapsed to form vertex $z \in V_{i+1}$, the total weight of vertices u and v become the weight of z, the edges incident on z is set equal to the union of the edges incident on u and v minus the edge (u, v). If there is an edge that is incident on both u and v, then the weight of this edge is set equal to the sum of the weights of these two edges. Vertices that are not incident on any edge of the matching are copied to G_{i+1}. In the maximal matching, vertices which are not neighbors are searched. In HEM, the vertices are visited in random order, but the collapsing is performed with the vertex that has the heaviest weight edge with the chosen vertex. Vertices are visited in random order and an adjacent vertice is chosen in random order as well in RM. During the succesive coarsening phases, the weights of vertices and edges increase. The coarsest graph can then be partitioned and further refinements can be achieved by suitable algorithms like Kernighen and Lin [7]. Finally, the partition of the coarsest graph is iteratively reformed back to the original graph by going through the graphs G_{k-1}, G_{k-2}, ...,G_1.

For the routing protocol, we propose a partitioning method called *Fixed Centered Partitioning* (FCP) where several fixed centers are chosen and the graph is then coarsened around these fixed centers by collapsing the heaviest edges around them iteratively. Different than [6], FCP does not have a matching phase, therefore iterations are much faster. Fig. 1 shows an example where a regular graph of 10 nodes is partitioned into three clusters. The initial fixed centers are encircled and the first collapsed neighbors are shown. The collapsing phases and the nodes collapsed at each iteration are depicted in Fig. 1.b and Fig. 1.c. The final partition has 3 partitions and a total cost of 13 for inter-partition edges. One problem with FCP is the initial allocation of the fixed centers. One possible solution is to choose the fixed centers randomly so that they are all at least some bounded distance from each other. The heuristic for the bound we used is $h = 2d \ / \ p$ where d is the diameter of the network and p is the number of partitions (clusters) to be formed.

Lemma 1. *FCP performs partitioning of $G(V, E)$ in $O(\lfloor n/p \rfloor)$ steps where $|V| = n$ and p is the number of clusters (partitions) required. The time complexity of the total collapsing of FCP is $O(n)$.*

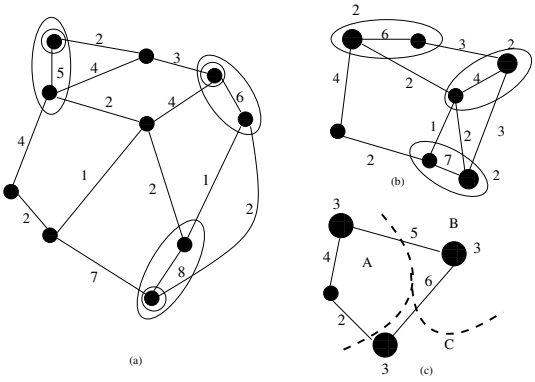


Fig. 1. Fixed Centered Partitioning

Proof. The FCP simply collapses p nodes with its heaviest edges at each step resulting in n/p steps. Since there are p collapsing at each step, total time complexity is $O(n)$.

3 The Hierarchical Routing Protocol

In the hierarchical routing protocol called the *Neighbor Protocol* (NP), each cluster has a representative neighbor node and one of the representatives is the coordinator. The distribution of the connectivity information is from the coordinator to the representatives and then from the representatives to the individual nodes as in a tree structure. When an ordinary mobile node changes its position, it sends a ND_TOP message to inform its representative that it now has different coordinates and goes into a WAIT state to wait until a new connectivity message (ND_ROUTE) is received from the representative as shown in Fig.2. The representative collects the ND_TOP messages until a timeout and then sends all of the current connectivity information to the coordinator in a REP_CONFIG message. The coordinator, after collecting the REP_CONFG messages within its timeout, starts the partitioning process of the updated network graph.

The coordinator concludes this step by identifying the nodes and neighbors in each cluster. It identifies one of the neighbor nodes as the representative for each cluster and sends the connectivity matrix (REP_PART) to the representatives. The representative for each cluster then distributes the local cluster connectivity information (ND_ROUTE) to individual nodes in its cluster in parallel with the other clusters. The representative also distributes the neighbor information to the neighbor nodes in its cluster. Ordinary nodes perform APSP to find their local shortest routes. Neighbor nodes however, perform APSP for intra-cluster and then inter-cluster routes.

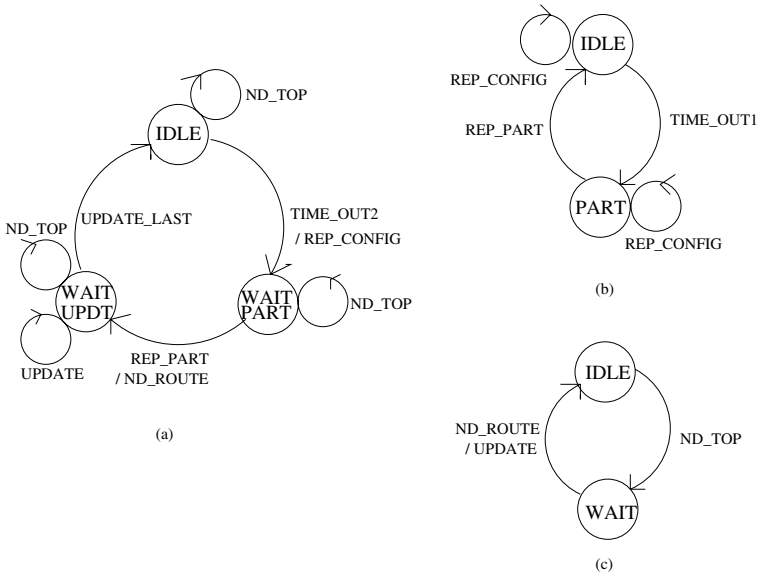


Fig. 2. The FSM Diagrams of Representative (a) , Coordinator (b) and Node (c)

4 An Example Network and Evaluations

An example network is depicted in Fig. 3. The initial centers allocated are 2, 11, 20 and 29. The coordinator and also the representative for cluster C is at 24. The coordinator partitions the graph using FCP as in Table. 1.

Table 1. Partitioning of the Example Network by FCP

A	B	C	D
G_0 — 2	11	20	29
G_1 — 2-1	11-12	20-19	29-28
G_2 — 2-1-5	11-12-8	20-19-17	29-28-25
G_3 — 2-1-5-4	11-12-8-14	20-19-17-22	29-28-25-26
G_4 — 2-1-5-4-3	11-12-8-14-15	20-19-17-22-21	29-28-25-26-23
G_5 — 2-1-5-4-3-6	11-12-8-14-15-13	20-19-17-22-21-18	29-28-25-26-23-27
G_5 — 2-1-5-4-3-6-7	11-12-8-14-15-13-10	20-19-17-22-21-18-16	29-28-25-26-23-27-24

Based on the partitioning information, the representatives chosen from the neighbors as 7, 10 and 17 are informed of their local connection. In the second phase, the representatives transfer this information to local nodes in their clusters in parallel. The ordinary nodes then calculate APSP in parallel, however, the neighbor nodes have to also calculate APSP for the simplified network graph which consists of the neighbor nodes only as shown in Fig. 4.

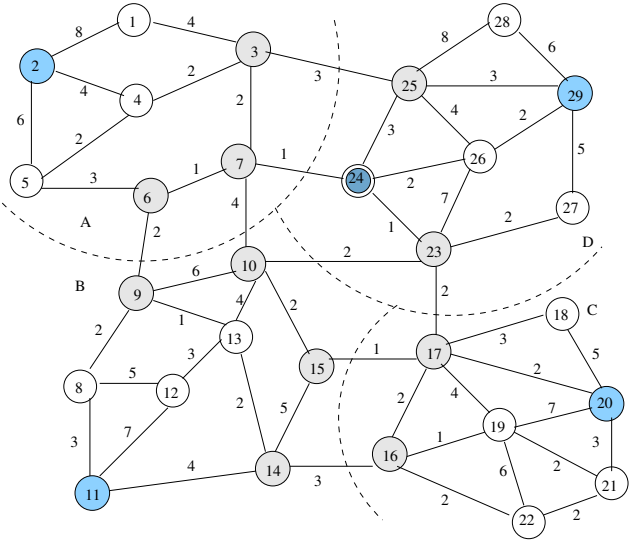


Fig. 3. The Original Network

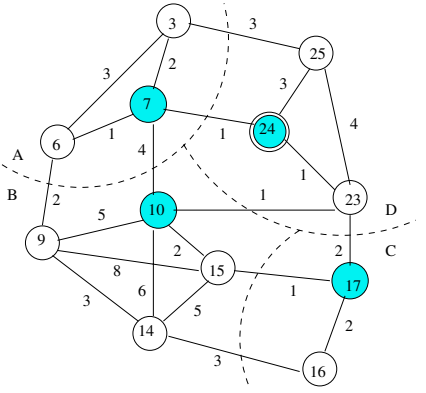


Fig. 4. The Simplified Neighbor Network

Consider an example where node 5 in cluster A wants to send a message to the node 22 in cluster C . Since destination is not in its own cluster, 5 sends the message to its closest neighbor node, 6. Node 6 sends the message to node 17 which is its closest neighbor node in cluster C over 6-7-24-23-17. The neighbor node 17 routes the message to the destination over the shortest path which is 17-16-22. The total cost of this path using NP or APSP is 12. The routes found by the method we proposed and any APSP algorithm such as Dijkstra's are compared for the example network for any pair of nodes that are in different clusters. The average coincidence for all pairs of nodes in each cluster for this example network, 85 % of the NP routes are coincident with APSP for a total of 308 routes.

5 Analysis

The performance analysis should include the following

1. Partitioning of the network graph by FCP : O_1
2. Distribution of the cluster connectivity messages to the cluster representatives : O_2
3. Distribution of the routing information to the individual nodes by each representative : O_3
4. Intra-cluster route calculation time by the nodes within the cluster : O_4
5. Inter-neighbor route calculation by the neighbors : O_5

Lemma 2. *Distribution of individual cluster routing information to the nodes (steps 2 and 3 above) take $O_{dist}(m)$ time where m is an upper bound on the number of nodes in a cluster*

Proof. The coordinator sends the cluster connections and the neighbor identities to all of the representatives in $\Theta_2(p)$ steps where p is the number of clusters in the network. Similarly, the representatives transfer this information to the individual nodes in $\Theta_3(m)$ steps in parallel with the other representatives. Assuming $m \gg p$, total time taken is $O_{dist}(m)$.

Lemma 3. *The total time required for intra-cluster and inter-neighbor routing algorithms is $O_{route}(m^3)$*

Proof. Since each node performs all-to-all routing in parallel, time spent for finding intra cluster routes is $O_4(m^3)$. Similarly, time spent for inter-neighbor shortest paths is $O_5(k^3)$ where k is an upper bund on the number of neighbors. Assuming $m \gg k$, total time is dominated by $O_{route}(m^3)$.

Theorem 1. *The Speedup obtained by the proposed protocol to a pure sequential all-to-all shortest paths protocol is $O(p^3)$ and to the parallel case where each node calculates all of the routes in parallel with others is $O(p^2/m)$.*

Proof. Total time for the protocol (O_{prot}) by Lemmas 1-3 is :

$$O_{prot} = O_{part}(n) + O_{dist}(m) + O_{route}(m^3) = O(n + m^3) \tag{1}$$

and assuming a balanced partition, that is, $n = mp$

$$O_{prot} = O(n + m^3) = O(mp + m^3) \tag{2}$$

Assuming the network has p clusters and m nodes at each cluster, a serial algorithm to compute all routes of this network will take $O_{serial}((p * m)^3)$ operations. The speedup S that can be approximated with respect to pure serial case is :

$$S = O_{serial}/O_{prot} = O((p * m)^3/(mp + m^3)) \tag{3}$$

and assuming $m \gg p$

$$S = O(p^3) \tag{4}$$

For the pure parallel case where each node has all of the network connectivity information, $O_{par} = O_(p^2m^2)$ and the speedup now is :

$$S = O(p^2m^2/m^3) = O(p^2/m) \tag{5}$$

This result may be interpreted as the more partitions the network has, the more speedup we obtain as $S = O(p)$ when $p = m$. This is not necessarily true as the number of partitions increases, the average nodes in a partition (m) would decrease for a given network which means that the assumption made about the relative magnitudes of p and m $(m \gg p)$ will not hold.

6 Experimental Results

The results for the partitioning of the graphs are shown in the figures below for the four algorithms as FCP, RFCP, RM and HEM where centers are initially allocated at random in RFCP. In Fig. 5, the average edge costs on sample graphs by the four algorithms are plotted. FCP and RFCP provide less total edge costs than the other two algorithms as shown.

The partition quality of the FCP, RM and HEM algorithms are shown in Fig. 6 for 10000 nodes where FCP and RFCP both have significant improvements over RM and HEM as expected since FCP partitions the graph into almost equal partitions as stated in Lemma 1.

For the static routing tests shown in Fig. 7, the shortest routes were first calculated by APSP algorithm for various random sized graphs. Then, the graphs were partitioned into clusters, the neighbor graphs were constructed and the shortest paths were calculated using the Neighbor Protocol. The latter routes

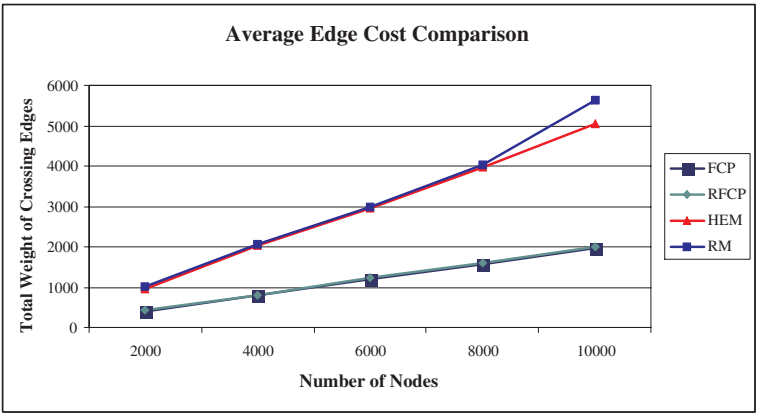


Fig. 5. Edge Cost Comparison

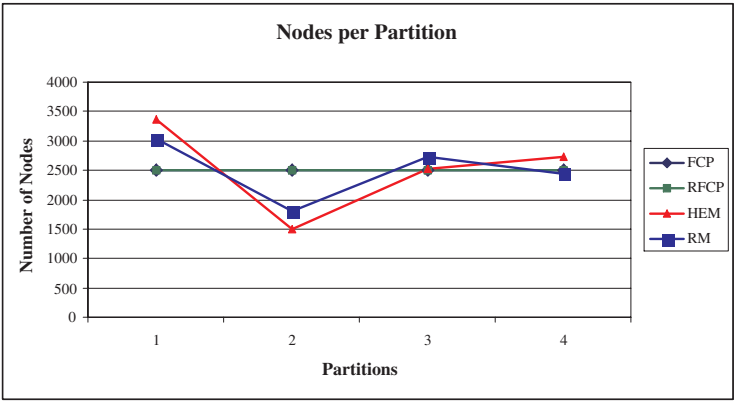


Fig. 6. Partition Quality Comparison

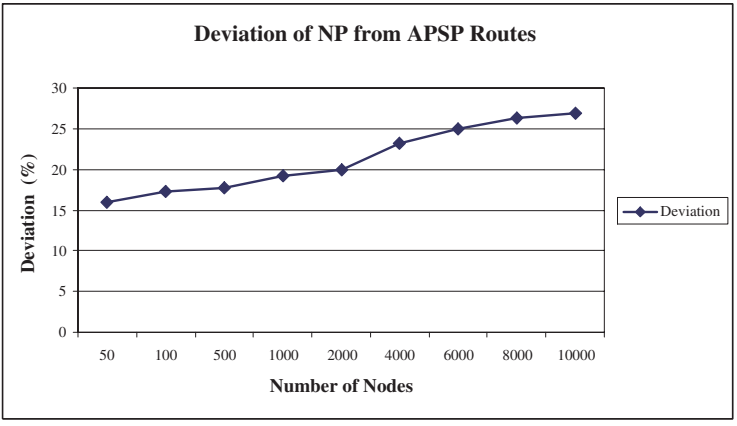


Fig. 7. Deviation of the Neighbor Protocol from APSP

are compared to the APSP routes and their deviations are calculated. We see that NP has about 27 % deviation from APSP in the largest graph set under consideration which has about 10000 nodes.

7 Conclusions

We proposed a dynamic routing protocol for a mobile network called the Neighbor Protocol, provided a graph partitioning heuristic and a hierarchical model to perform routing in parallel. We showed that this approach improves performance considerably theoretically. However, although realistic, the speedup obtained in Section 5 is optimistic in a sense because of the few assumptions made about the number of clusters and the nodes in a cluster. The method we propose for

routing in mobile networks provides *good* routes which are not necessarily the shortest paths but are comparable to shortest paths as shown by the tests. We are planning to evaluate the performance of NP in terms of total control traffic against the frequency of route requests and frequency of movement in a mobile network using simulations and comparing these with other mobile network routing protocols such as zone routing and k-way clustering.

We are also looking into the fully distributed version of this protocol for mobile ad hoc networks for two cases. In the first case, there is no central coordinator but there are representatives and decisions on the partitioning of the graph and routing are done at the representative level by distributed agreement. In the second case, there are no central components which would require fully distributed algorithms.

Acknowledgements. We wish to thank Pinar Dűndar of Ege University, Dept. of Math. for her valuable discussions of the graph partitioning heuristics.

References

1. Busch, C. et al.: Analysis of Link Reversal Routing Algorithms for Mobile Ad hoc Networks. Proc. of the 15th ACM Symp. on Parallel Alg. and Arch. (2003) 210–219
2. E.W. Dijkstra: A Note on Two Problems in Connection with Graphs. Numerische Math., Vol. 1. (1959) 69–271
3. Fernandess,Y., Malki, D.: K-clustering in Wireless Ad hoc Networks, Proc. of the second ACM Int. Workshop on Principles of Mobile Computing, (2002) 31–37
4. R.W. Floyd: Algorithm 97 (Shortest path), CACM, Vol. 5(6). (1962) 345
5. Z.J. Haas, M.R. Pearlman: The Zone Routing Protocol (ZRP) for Ad hoc Networks. Internet Draft, Internet Engineering Task Force. (1997)
6. Karypis, G., Kumar V.: Multilevel k-way Partitioning scheme for Irregular Graphs. Journal of Parallel and Distributed Computing, Vol. 48. (1998) 96–129
7. Kernighan, B., Lin, S.: An Effective Heuristic Procedure for Partitioning Graphs. The Bell System Technical Journal, (1970) 291–308
8. P. Krishna et al.: A Cluster-based Approach for Routing in Dynamic Networks. ACM SIGCOMM Comp. Comm. Rev., Vol. 27(2). (1997) 49–64
9. V.D. Park, M.S. Corson: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. Proc. IEEE INFOCOM, Vol. 3. (1997) 1405–1413

Distributed Optimization of Fiber Optic Network Layout Using MATLAB

Roman Pfarrhofer[1], Markus Kelz[1], Peter Bachhiesl[1], Herbert Stögner[1], and Andreas Uhl[1,2]

[1] Carinthia Tech Institute, School of Telematics & Network Engineering
Primoschgasse 8, A-9020 Klagenfurt, Austria
[2] Salzburg University, Department of Scientific Computing
Jakob-Haringerstr.2, A-5020 Salzburg, Austria

Abstract. A MATLAB-based planning tool for the computation of cost optimized laying for fiber optic access networks is employed on homogenous and heterogenous Windows PC networks. The approach is based on a custom MATLAB toolbox which does not require a MATLAB client installed at the participating machines. Experiments demonstrate the efficiency and real-world usability of the approach which facilitates an interactive network planning process.

1 Introduction

The demand for broadband capacities has entailed an increasing changeover from conventional transmission techniques to the fiber optic technology [1]. During the last two years European network-carriers have invested 7.5 billion Euros in the expansion of the core and the distribution net domain (backbones, city backbones and metropolitan area networks). However, investigations have shown that about 95% of the total costs for the implementation may be expected for the area-wide realization of the last mile (access networks). In order to achieve a return on investment, carriers will be forced to link access net nodes, like corporate clients, private customers or communication nodes for modern mobile services (e.g. UMTS) to their city backbones.

As a consequence, the cost optimized laying of underground networks (e.g. fiber optical networks) with respect to the given spatial topologies and under consideration of usable infrastructures represents a challenging task of practical interest. Whereas a variety of strategic planning tools focus on problems of laying optimization, discussion of reliability, and traffic optimization in the core net domain [2], carriers hardly use any computational planning tools for cost estimation and optimization in the access net domain. The planning process is performed manually based on expert knowledge and therefore often yield suboptimal decisions. However, since the telecommunication market runs through a period of consolidation the neglection of economization potentials has to be considered as a critical competitive disadvantage.

In recent work [3], we have introduced the methodologies behind the MATLAB-based planning tool NETQUEST-OPT for the computation of cost

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 538–547, 2004.
© Springer-Verlag Berlin Heidelberg 2004

optimized and real world laying for fiber optic access networks. Real-world topological geometries are represented by detailed geoinformation data. The optimization kernel is based on cluster strategies, exact and approximative graph theoretical algorithms, combinatorial optimization, and ring closure heuristics. It includes subproblems which are polynomially or even exponentially growing with respect to the complexity of the underlying geometries. As a consequence, NETQUEST-OPT currently requires computation times of several hours on a single workstation even for medium sized access domains. For an efficient planning process interactivity is a desired property, e.g. for rapid prototyping of different clustering strategies. This goal can only be achieved by using high performance computing systems.

In this work, we focus on the efficient use of NETQUEST-OPT in a distributed environment using a custom distributed MATLAB approach based on the PARMATLAB and TCPIP toolboxes. In Section 2, we describe previous approaches to use MATLAB in parallel and distributed environments and shortly review our own development MDICE. Section 3 describes the basic principles and the structure of NETQUEST-OPT and subsequently discusses a computing intensive part of the tool in some detail. In Section 4 we present experimental results of execution in homogenous and heterogenous environments.

2 MATLAB in Parallel and Distributed Computing

MATLAB has established itself as the numerical computing environment of choice on uniprocessors for a large number of engineers and scientists.

For many scientific applications, the desired levels of performance are only obtainable on parallel or distributed computing platforms. Therefore, a lot of work has been done, both in industry and in academia, to develop high performance MATLAB computing environments.

C. Moler stated in "Why there isn't a parallel MATLAB" (Mathworks newsletter in 1995) that due to the lack of widespread availability of high performance computing (HPC) systems there would be no demand for parallel MATLAB. With the emerge of cluster computing and the potential availability of HPC systems in many universities and companies, the demand for such a software system is obvious. A comprehensive and up-to-date overview on high performance MATLAB systems is given by the "Parallel MATLAB Survey"[1]. Several systems may be downloaded from the MATHWORKS FTP[2] and WWW[3] servers. There are basically three distinct ways to use MATLAB on HPC architectures:

1. Developing a high performance interpreter
   a) Message passing: communication routines usually based on MPI or PVM are provided. These systems normally require users to add parallel instructions to MATLAB code [4,5,6].

[1] http://supertech.lcs.mit.edu/~cly/survey.html
[2] ftp://ftp.mathworks.com/pub/contrib/v5/tools/
[3] http://www.mathtools.net/MATLAB/Parallel/

   b) "Embarrassingly parallel": routines to split up work among multiple MATLAB sessions are provided in order to support coarse grained parallelization. Note that the PARMATLAB and TCPIP toolboxes and our own development MDICE [7] fall under this category.

2. Calling high performance numerical libraries: parallelizing libraries like e.g. SCALAPACK are called by the MATLAB code [8]. Note that parallelism is restricted within the library and higher level parallelism present at algorithm level cannot be exploited with this approach.
3. Compiling MATLAB to another language (e.g. C, HPF) which executes on HPC systems: the idea is to compile MATLAB scripts to native parallel code [9,10,11]. This approach often suffers from complex type/shape analysis issues

Note that using a high performance interpreter usually requires multiple MATLAB clients whereas the use of numerical libraries only requires one MATLAB client. The compiling approach often does not require even a single MATLAB client. On the other hand, the use of numerical libraries and compiling to native parallel code is often restricted to dedicated parallel architectures like multicomputers or multiprocessors, whereas high performance interpreters can be easily used in any kind of HPC environment. This situation also motivates the development of our custom high performance MATLAB environment: since our target HPC systems are (heterogenous) PC clusters running a Windows system based on the NT architecture, we are restricted to the high performance interpreter approach. However, running a MATLAB client on each PC is expensive in terms of licensing fees and computational resources. Consequently, our aim was to develop a high performance interpreter which requires one MATLAB client for distributed execution only.

MATLAB **DI**stributed **C**omputing **E**nvironment (MDICE [7]) is based on the PARMATLAB and TCPIP toolboxes. The PARMATLAB toolbox supports coarse grained parallelization and distributes processes among MATLAB clients over the intranet/internet. Note that each of these clients must be running a MATLAB daemon to be accessed. The communication within PARMATLAB is performed via the TCPIP toolbox. Both toolboxes may be accessed at the Mathworks ftp-server (referenced in the last Section) in the directories `parmatlab` and `tcpip`, respectively.

However, in order to meet the goal to get along with a single MATLAB client the PARMATLAB toolbox needed to be significantly revised. The main idea was to change the client in a way that it can be compiled to a standalone application. At the server, jobs are created and the solve routine is compiled to a program library (*.dll). The program library and the datasets for the job are sent to the client. The client is running as background service on a computer with low priority. For this reason the involved client machines may be used as desktop machines by other users during the computation (however, this causes the need for a dynamic load balancing approach of course). This client calls over a predefined standard routine the program library with the variables sent by the

server and sends the output of the routine back to the server. After the receipt of all solutions the server defragments them to an overall solution.

For details on MDICE and corresponding performance results for a Monte Carlo simulation application see [7].

3 Optimization of Fiber Optic Network Layout in the Access Net Domain

3.1 Sequential Approach

In order to consider real world conditions, detailed geoinformation data, usually supplied by the network carriers, are essential. We map the real world geometries to a set of nodes V of a graph $G = (V, E)$. In the simplest case the optimization of network layings means to find an optimal subset E^* of the set of edges E such that all network users $R \subset V$ (R is the set of access nodes) are reachable.

According to the methods in [3] the following additional information is extracted from the GIS databases:

Penalty Grid: A spatially balanced score card combines all relevant land classes with typical implementation costs. The access net domain is regarded as a regular grid $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, i and j are indices of finite index sets. Each entry of the penalty matrix P describes the specific implementation costs for the grid pixel $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$.

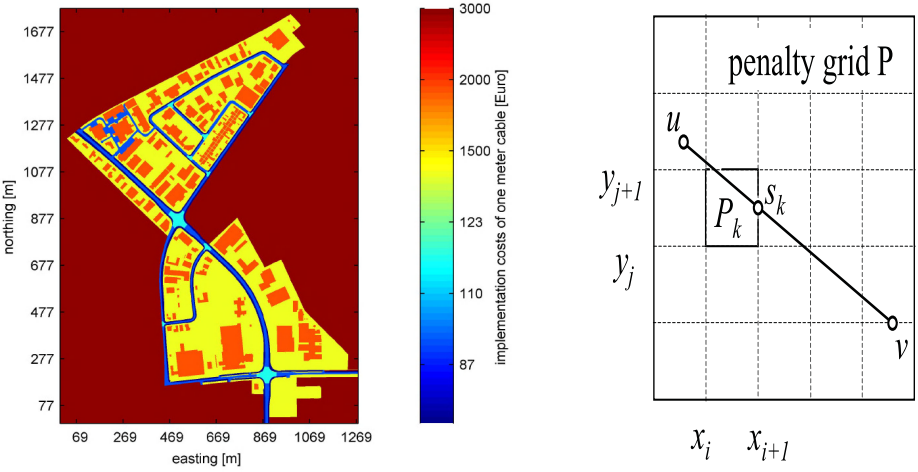
Auxiliary geometries: Plane geometries of all relevant land classes are exported. The shapes of these geometries are determined by a set S of auxiliary nodes.

For a detailed review of some elaborate laying problems treated by NETQUEST-OPT (e.g. consideration of redundancy requirements) we refer to [3]. In all cases of optimization we have to construct a weighted graph $G = (V, E; W)$. Thus we penalize the edges in E with respect to the underlying penalty grid P for the considered network domain. Fig. 1.a shows the corresponding implementation costs for the different land classes of the penalty grid. Figs. 1 and 2 correspond to an industrial benchmark project based on data provided by NetCologne, one of the major private city net carriers in Germany.

For an edge $[u, v] \in E$, $u \in V$, $v \in V$, $u \neq v$, we define the entry $W_{u,v}$ of the symmetric cost matrix $W \in Mat\,(|V| \times |V|)$ according to

$$W_{u,v} := \sum_{k=0}^{K-1} \|s_{k+1} - s_k\|_2 \, P_{k+1} \ . \tag{1}$$

As depicted in Figure 1.b, s_k denotes the (x, y) -coordinates of the k-th intersection point of the edge $[u, v]$ with the grid lines of the cost grid P. P_k represents the specific implementation costs corresponding to the predecessor pixel for the k-th intersection point s_k of an edge $[u, v]$ with the grid lines of the cost grid. The predecessor notation corresponds to the direction from u to v.



(a) Implementation costs

(b) Derivation of the costs

Fig. 1. Penalty grid in the context of network laying

The computation of W is one of the most time demanding procedures in the entire laying optimization process and requires $(|V|-1)\,|V|\,/2$ calls of (1). In real world geometries, the network planning problem leads to dimensions up to $|V| \cong 20000$ nodes. Due to memory restrictions on our target architectures, these problem dimensions required us to segment the entire optimization procedure into three hierarchical steps: determination of local clusters, local solution in each cluster and computing cluster shortest spanning trees [3].

Fig. 2.a shows the required access nodes for the domain shown in Fig. 2.b. The problem is partitioned into three clusters. The black circles depict the auxiliary nodes of the first cluster based on the underlying geometries, gridding is done with a resolution of one meter in square.

The solution depicted in Fig. 2.b was computed in 223 minutes on three PCs. The result saves 22% distance and 20% costs with respect to the original NetCologne laying.

3.2 Distribution Strategy

The segmentation of the spatial domain into independent clusters for an efficient solution of the network laying problem as described above allows also straightforward and computationally efficient distributed processing at first hand. However, this clustering procedure (inter-cluster parallelism) leads to suboptimal global umbrella networks only, especially in case of a larger number of clusters which would be required for scalable parallel or distributed execution. Therefore, clustering must not be used as a means to provide parallelism. For this reason, we

(a) 37 access nodes (triangles) and three clusters

(b) Arial view of target area with overlayed computed solution

Fig. 2. Required access nodes and computed solution.

propose a "intra-cluster parallelism" approach which does not exploit the parallelism introduced by the clustering process at all. As a consequence, an optimal global solution may be obtained if clustering can be avoided (i.e. the target architecture allows a single-cluster approach).

The computations associated with the evaluation of the cost matrix W (which are covered by our experiments) may be distributed in a simple fashion by domain decomposition without the requirement of inter-process communication. This simple partitioning approach leads to entirely deterministic behaviour since the computational effort of evaluating parts of the matrix W is not data dependent. Therefore, load balancing functionalities seem not to be necessary at first hand. However, due to the possible interference of other applications (recall that the clients run with low priority on desktop machines) and the potential employment of the application in heterogeneous environments, we use a simple load balancing approach (see below). We emphasize that MDICE is also employed for other, more sophisticated subproblems of NETQUEST-OPT – we exemplarily mention the solution of Steiner Network Problems [12], the solution of cluster interconnecting minimal spanning tree problems or the computation of ring closure matrices for redundant networks [3]. However, due to the clarity of the results with respect to the use of MDICE we cover the evaluation of W as exemplary part of NETQUEST-OPT.

4 Experiments

4.1 Experimental Settings

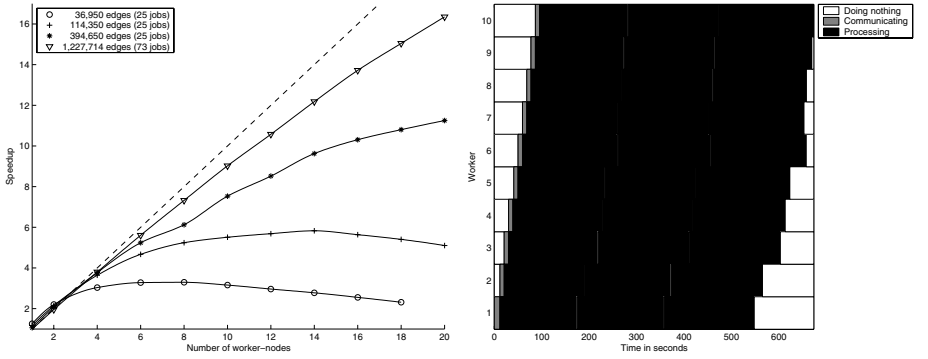
The data required for computing the cost matrix W is split into a certain number of equally sized jobs N to be distributed among the clients M by the server ($N \geq M$). Whenever a client has sent back its result to the server after the initial distribution of M jobs to M clients, the server assigns another job to this idle client until all N jobs are computed. This approach is denoted "asynchronous single task pool method" [13]. In case $N \gg M$ this strategy is a dynamic load balancing approach which can also cope with heterogeneous environments. When the server contacts a client machine the first time, the compiled client software is sent in addition to the required data to perform the first computational job. For all subsequent jobs to be executed on this machine, only data has to be sent (which is much less demanding of course). Note that since MDICE does not provide tools for automatic parallelization, the N jobs (i.e. the decomposition of W) need to be specified before the entire computation starts.

We consider cost matrices W from real-world problems associated with a different number of edges (i.e. 36950, 114350, 394650, and 1227714) in order to show the impact of varying the problemsize on the performance of our solution. Additionally, we vary the number of jobs distributed among the worker clients after fixing the problemsize to 394650 edges to show the tradeoff between good load distribution versus communication effort. The computing infrastructure consists of the server machine (1.99 GHz Intel Pentium 4, 504 MB RAM, Windows XP Prof.) and two types of client machines (996 MHz Intel Pentium 3, 128 MB RAM, and 730 MHz Intel Pentium 3, 128 MB RAM, both types under Windows XP Prof.). The Network is 100 MBit/s Ethernet. In order to demonstrate the flexibility of our approach, we present results in "homogenous" and "heterogenous" environments. In the case of the homogenous environment, we use client machines of the faster type only, the results of the heterogenous environment correspond to six 996 MHz and four 730 MHz clients, respectively. Note that the sequential reference execution time used to compute speedup (see Fig. 3.a) has been achieved on a 996 MHz client machine with a compiled (not interpreted) version of the application to allow a fair comparison since the client code is compiled as well in the distributed application. We use MATLAB 6.5.0 with the MATLAB compiler 3.0 and the LCC C compiler 2.4.

4.2 Experimental Results

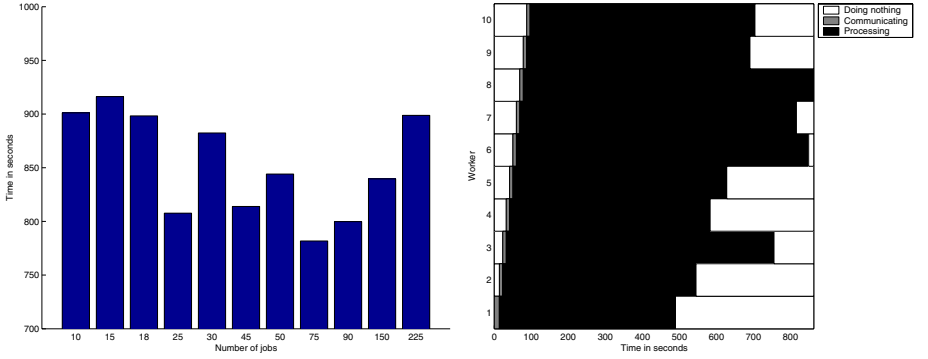
First we discuss the homogenous environment. In Fig. 3.a we display speedup results for differently sized matrices W. It is clearly exhibited that speedup saturates for the smaller problems at 4 or 8 clients, respectively. On the other hand, reasonable scalability is shown for the larger problems up to 20 clients.

Fig. 3.b shows a visualization of the distributed execution, where black areas represent computation time-intervals and gray areas communication events. The number of edges has been fixed to 394650, the number of clients is set to 10,



(a) Speedup for varying problem-size (b) Visualization of execution

Fig. 3. Results in the homogenous environment.

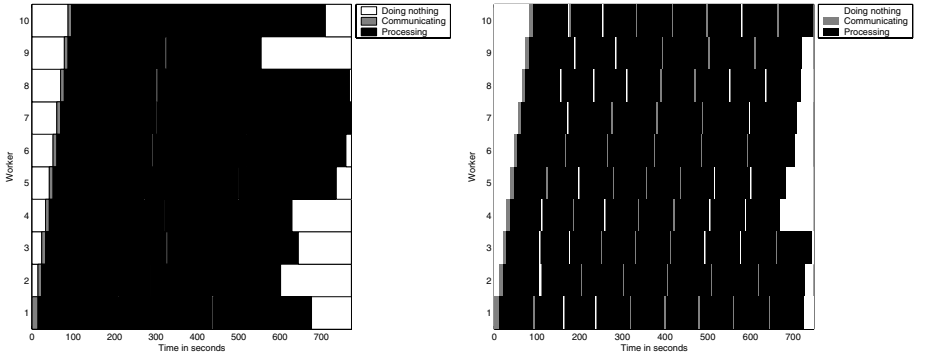


(a) Execution time with a varying number of jobs (b) Visualization of execution with 10 jobs

Fig. 4. Results in the heterogenous environment.

and 30 jobs are used. We clearly notice the staggered start of computation at different clients which is due to the expensive initial communication phase, where the server has to send the compiled code and input data to each of the clients. This behaviour is the reason why the settings $N = M$ or $N = n \times M$ (with small n, where $n = 3$ in our case) do not perform as well as expected. The solution is to increase the number of jobs in order to balance the load at the end of the application, however, the delayed start at several clients can not be avoided.

In the following we discuss the heterogenous environment, again the number of edges has been fixed to 394650 and the number of clients is set to 10 as specified in the last Section. In Fig. 4.a we vary the number of jobs distributed among the clients. Clearly, a high number of jobs leads to an excellent load distribution,



(a) 25 jobs (b) 75 jobs

Fig. 5. Visualization of execution behaviour in the heterogeneous environment.

but on the other hand the communication effort is increased thereby reducing the overall efficiency.

For our target system, the optimal number of jobs is identified to be 75. A further increase leads to an increase of execution time. Fig. 4.b shows an execution visualization where $N = M = 10$. Of course, the performance is worse as compared to the analogous homogenous case and the slower clients (numbers 3,6,7,8) are immediately identified.

Finally, in Figs. 5.a and 5.b we show the execution behaviour for $N = 25$ and $N = 75$, respectively. Although the structure and the visual appearance of the two cases is quite different, we result in almost identical execution times. Whereas for $N = 25$ we have little communication effort but highly unbalanced load, the opposite is true for $N = 75$. The tradeoff between communication and load distribution is inherent in the single pool of task approach which means that the optimal configuration needs to be found for each target environment.

5 Conclusion

In this work, we employ a MATLAB-based planning tool for the computation of cost optimized laying for fiber optic access networks on homogenous and heterogenous Windows PC networks. The custom MATLAB toolbox MDICE may take advantage of the large number of Windows NT-architecture based machines available in companies and universities. It proves to be efficient in terms of low licencing fees and satisfying execution behaviour. The distributed approach suggested in this work allows for an interactive network planning process provided enough PCs are available. Additionally, our approach may lead to a better quality of the computed network solution as compared to a sequential clustering approach in case the clustering can be avoided in the distributed execution.

References

[1] G. Gilder. *Telecosm: The world after bandwidth abundance.* Touchstone Books, Charmichael, 2002.

[2] D. Bertsekas. *Network Optimization: Continous and Discrete Models.* Athena Scientific, Belmont, 1998.

[3] P. Bachhiesl, G. Paulus, M. Prossegger, J. Werner, and H. Stögner. Cost optimized layout of fibre optic networks in the access net domain. In U. Leopold-Wildburger, F.Rendl, and G. Wäscher, editors, *Proceedings of Operations Research OR'02*, Springer Series on Operations Research, pages 229–234. Springer Verlag, 2002.

[4] J.F. Baldomero. PVMTB: Parallel Virtual Machine Toolbox. In S. Dormido, editor, *Proceedings of II Congreso de Usarios MATLAB'99*, pages 523–532. UNED, Madrid, Spain, 1999.

[5] V.S. Menon and A.E. Trefethen. MultiMATLAB: integrating MATLAB with high performance parallel computing. In *Proceedings of 11th ACM International Confernce on Supercomputing.* ACM SIGARCH and IEEE Computer Society, 1997.

[6] S. Pawletta, T. Pawletta, and W. Drewelow. Comparison of parallel simulation techniques - MATLAB/PSI. *Simulation News Europe*, 13:38–39, 1995.

[7] R. Pfarrhofer, P. Bachhiesl, M. Kelz, H. Stögner, and A. Uhl. MDICE – a MATLAB toolbox for efficient cluster computing. In *Proceedings of Parallel Computing 2003 (ParCo 2003).* Elsevier Science B.V., 2004. To appear.

[8] S. Ramaswamy, E.W. Hodges, and P. Banerjee. Compiling MATLAB programs to SCALAPACK: Exploiting task and data parallelism. In *Proceedings of the International Parallel Processing Symposium (IPPS)*, pages 814–821. IEEE Computer Society Press, 1996.

[9] L. DeRose and D. Padua. A MATLAB to Fortran 90 translator and its effectiveness. In *Proceedings of 10th ACM International Conference on Supercomputing.* ACM SIGARCH and IEEE Computer Society, 1996.

[10] P. Drakenberg, P. Jakobson, and B. Kagström. A CONLAB compiler for a distributed memory multicomputer. In *Proceedings of the 6th SIAM Conference on Parallel Processing for Scientific Computing*, volume 2, pages 814–821, 1993.

[11] M. Quinn, A. Malishevsky, N. Seelam, and Y. Zhao. Preliminary results from a parallel MATLAB compiler. In *Proceedings of the International Parallel Processing Symposium (IPPS)*, pages 81–87. IEEE Computer Society Press, 1998.

[12] A.Z. Zelikovsky. An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.

[13] A.R. Krommer and C.W. Überhuber. Dynamic load balancing – an overview. Technical Report ACPC/TR92-18, Austrian Center for Parallel Computation, 1992.

Cache Conscious Dynamic Transaction Routing in a Shared Disks Cluster

Kyungoh Ohn and Haengrae Cho

Department of Computer Engineering, Yeungnam University
Kyungsan, Kyungbuk 712-749, Republic of Korea
{ondal,hrcho}@yu.ac.kr

Abstract. A shared disks (SD) cluster couples multiple computing nodes for high performance transaction processing, and all nodes share a common database at the disk level. In this paper, we propose a transaction routing algorithm employed at the front-end router to select an execution node of an incoming transaction. The proposed algorithm improves the system performance by increasing the buffer hit ratio and reducing the frequency of cross buffer invalidation while achieving the dynamic load balancing. Using a simulation model, we evaluate the performance of the proposed algorithm under a wide variety of database workloads.

1 Introduction

A cluster is a collection of interconnected computing nodes that collaborate on executing an application. Depending on the nature of disk access, there are two primary flavors of cluster architecture designs: *shared nothing* (SN) and *shared disks* (SD) [14,15]. In the SN cluster, each node has its own set of private disks and only the owner node can directly read and write its disks. On the other hand, the SD cluster allows each node to have direct access to all disks. The SD cluster offers a number of advantages compared to the SN cluster, such as dynamic load balancing and seamless integration, which make it attractive for high performance transaction processing. Furthermore, the rapidly emerging technology of storage area networks (SAN) makes SD cluster the preferred choice for reasons of higher system availability and flexible data access. The recent parallel database systems using the SD cluster include IBM DB2 Parallel Edition [5] and Oracle9*i* Real Application Cluster [10,12].

Each node in the SD cluster has its own buffer pool and caches database pages in the buffer. Caching may substantially reduce the number of disk I/O operations by utilizing the locality of reference. However, since a particular page may be simultaneously cached in different nodes, modification of the page in any buffer invalidates copies of that page in other nodes. This necessitates the use of a *cache coherency scheme* so that the nodes always see the most recent version of database pages [2,3,4,8].

This paper proposes a transaction routing algorithm employed at the front-end router to select an execution node of an incoming transaction. In the SD

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 548–557, 2004.
© Springer-Verlag Berlin Heidelberg 2004

cluster, if transactions referencing similar data are clustered together to be executed on the same node (*affinity node*), then the buffer hit ratio should increase and the level of interference among nodes due to buffer invalidation will be reduced. This concept is referred to the *affinity-based routing* [9,14]. However the affinity-based routing is very much non-adaptive to the changes in the system load, that is, it does not take the current load distribution of each node into account while taking the routing decisions. This is particularly problematic when the load deviation of each node is quite large.

To avoid overloading individual node, a *dynamic load balancing* has to be considered. Unfortunately, supporting both affinity-based routing and dynamic load balancing are often contradictory goals [14]. This is because the dynamic load balancing distributes the congested transactions to multiple nodes, which may hurt the advantage of affinity-based routing. This paper proposes a new transaction routing algorithm in the SD cluster. Our main contributions are:

- We develop a dynamic transaction routing algorithm, named *Dynamic Affinity Cluster Allocation* (DACA). DACA is novel in the sense that it can make an optimal balance between the affinity-based routing and indiscriminate sharing of load in the SD cluster.
- We compare the performance of DACA with other transaction routing algorithms in the SD cluster under a wide variety of database workloads.

The rest of this paper is organized as follows. Sect. 2 summarizes the related work and Sect. 3 presents DACA in detail. Sect. 4 describes the simulation model and the experiment results. Finally, concluding remarks appear in Sect. 5.

2 Related Work

There are few studies on dynamic transaction routing in the SD cluster [9]. All of related studies select the execution node of incoming transactions based on data affinity when the system load is evenly balanced. However, there are some differences on handling overload due to the congestion of some transaction class. In [15], the incoming transactions of the congested class are spread across *all* nodes. On the other hand, in [6], the transactions are routed to the *least loaded* node. Both studies may achieve load balancing dynamically. However, the performance improvement could be marginal since the buffer hit ratio of each node decreases. The reduction comes from two factors [15]. First, in each of the nodes except the surge node (i.e., the node corresponded with the congested class) the granules in the buffer comes from two partitions: the original partition logically assigned to the node and the partition corresponding to the congested class. The second is the frequent cross invalidation effect on the granules of the partition corresponding to the congested class.

Dynamic transaction routing was studied extensively in the SN cluster [9]. However, the SN cluster limits inherently the potential for load balancing since the execution node of a database operation is *statically* determined by the physical database allocation. Specifically, for load balancing reasons, even though a

transaction is assigned to a lightly loaded load, a database operation referencing a remote database partition has to be shipped to the node owning the partition. Then the load of owner node is not significantly reduced.

The SD cluster also separates database operations into local and remote ones according to the required data item is cached in the local buffer or in the remote buffers (i.e., buffers of other nodes). However, it is important to note that the separation is determined *dynamically*. Once a node caches a data item in its local buffer, following operations on the data item can be processed locally. This offers the SD cluster much flexibility for load balancing, and requires a specific transaction routing algorithm for the SD cluster.

It is worthy to compare the transaction routing with the task scheduling in general distributed systems [1,7,11]. These studies often migrate a task to other node during its execution or decompose a task into many concurrent subtasks. We will not consider transaction migration in this paper. This is because most transactions execute in short duration [9]. The transaction decomposition will not be considered as well, since it requires a complex commitment protocol such as 2PC that is not adopted in most SD clusters.

3 Dynamic Affinity Cluster Allocation (DACA)

3.1 Preliminary

To alleviate the routing overhead, DACA considers to balance the load of each *affinity cluster* (AC) [14] not each transaction class. An AC collects several transaction classes with high affinity to a given set of relations. We assume that the number of ACs (#AC) is equal to or smaller than the number of nodes (#N) so as to minimize the load differences among ACs [14]. A *transaction router* maintains routing parameters and routes incoming transactions to nodes. Specifically, when a transaction router allocates a transaction of an affinity cluster AC_q to a node N_p, it increments both $\#\mathrm{T}(\mathrm{AC}_q)$ and $\#\mathrm{T}(N_p)$, which means the number of active transactions at AC_q and at N_p respectively. Both counters are decremented when a transaction commits or aborts.

The transaction router implements a *routing function*, $\mathcal{R}$, which specifies the set of nodes allocated for each AC. $|\mathcal{R}(\mathrm{AC}_q)|$ implies the number of nodes in $\mathcal{R}(\mathrm{AC}_q)$. $\mathcal{R}^{-1}$ is an inverse function of $\mathcal{R}$, and $|\mathcal{R}^{-1}(N_p)|$ implies the number of ACs in $\mathcal{R}^{-1}(N_p)$. If $\mathcal{R}(\mathrm{AC}_q)$ includes multiple nodes, incoming transactions of AC_q are routed to one of the nodes in a round-robin fashion. Initially, $\mathcal{R}(\mathrm{AC}_i)$ is set to $\{N_i\}$, which means that transactions of AC_i is routed to N_i. As a result, for every AC_q and N_p, $|\mathcal{R}(\mathrm{AC}_q)|$ is set to 1 and $|\mathcal{R}^{-1}(N_p)|$ may be 0 or 1 initially. It is possible to optimize the initial setting by assigning more nodes to some AC, if we expect that transactions of the AC will occupy large portion of the system load. Whenever a transaction is incoming or commits, the transaction router calculates an average load of a system $\bar{L}(N)$, which is $\Sigma_{i=1}^{\#N}(\#\mathrm{T}(N_i)\ /\ \#\mathrm{N})$.

DACA divides the type of overload into an *AC overload* and a *node overload*. The AC overload implies the case where transactions of some AC are congested. The node overload occurs when a node N_p ia allocated to several ACs and

$\#\mathrm{T}(N_p)$ is over average. In the followings, we define some terminologies to model the status of an AC and a node.

Definition 1 An affinity cluster AC_q is *overloaded*, if $\#\mathrm{T}(\mathrm{AC}_q)$ / $(|\mathcal{R}(\mathrm{AC}_q)| + 1) \geq \bar{L}(N)$.

Definition 2 A node N_p is *overloaded*, if $|\mathcal{R}^{-1}(N_p)| > 1$ and $\#\mathrm{T}(N_p) \geq \bar{L}(N) \times \alpha$, where α is a sensitivity factor and $1 < \alpha \leq 2$.

Definition 3 An affinity cluster AC_q is *underloaded*, if $|\mathcal{R}(\mathrm{AC}_q)| > 1$ and $\#\mathrm{T}(\mathrm{AC}_q)$ / $(|\mathcal{R}(\mathrm{AC}_q)| - 1) < \bar{L}(N)$.

DACA balances the load of each node according to the overload type. If AC_q is overloaded, then DACA allocates one more node to AC_q by expanding $\mathcal{R}(\mathrm{AC}_q)$. We refer this strategy as a *node expansion*. If there are no AC overloads but N_p is overloaded, then DACA distributes some ACs in $\mathcal{R}^{-1}(N_p)$ to other node. We refer this strategy as an *AC distribution*. Finally, if AC_q is underloaded, then DACA excludes some node from $\mathcal{R}(\mathrm{AC}_q)$. This strategy is referred as a *node reduction*. Now we describe each strategy in detail.

3.2 Node Expansion

Suppose AC_q is overloaded and $\mathcal{R}(\mathrm{AC}_q)$ is $\{N_q\}$. Since routing new transactions of AC_q to N_q increases the response time, the router expands $\mathcal{R}(\mathrm{AC}_q)$ to include additional node. We select the *least loaded* node, say N_k, as a candidate. This means that $\#\mathrm{T}(N_k)$ is the minimum among other nodes. Then $\mathcal{R}(\mathrm{AC}_q)$ is expanded to $\{N_q, N_k\}$, and incoming transactions of AC_q are routed to either N_q or N_k in a round-robin fashion.

A complicated case occurs when N_k has already been assigned to some affinity cluster, say AC_k. In this case, we first check whether AC_k is underloaded. If that is true, a node reduction strategy is applied so that $\mathcal{R}(\mathrm{AC}_k)$ excludes N_k, and then N_k is assigned to AC_q. We describe the node reduction strategy in Sect. 3.4. If AC_k is not underloaded, $\mathcal{R}(\mathrm{AC}_k)$ is changed to route transactions of AC_k to other node. The next section describes how to select a new node to execute transactions of AC_k. At both cases, AC_q becomes the only affinity cluster allocated to N_k.

Example 1: Suppose an example SD cluster with four nodes ($\#\mathrm{N} = 4$) and four ACs ($\#\mathrm{AC} = 4$). Initially, $\mathcal{R}(\mathrm{AC}_i)$ is set to $\{N_i\}$ for $1 \leq i \leq 4$, and for each AC, 50 transactions are in execution, i.e., $\#\mathrm{T}(\mathrm{AC}_i) = \#\mathrm{T}(\mathrm{N}_i) = 50$ for $1 \leq i \leq 4$. If the number of incoming transactions of AC_1 increases to 150, the average load of nodes $\bar{L}(N)$ becomes $\frac{150+50+50+50}{4} = 75$. In this case, AC_1 is overloaded since $\frac{150}{1+1} = 75 \geq \bar{L}(N)$. To resolve the overload state of AC_1, the router selects a least loaded node. Since the load of every node except N_1 is equal, the router may select any node as a candidate. Suppose that N_2 is selected as a candidate. Then $\mathcal{R}(\mathrm{AC}_1)$ becomes $\{N_1, N_2\}$ as a result of node expansion. $\mathcal{R}(\mathrm{AC}_2)$ can be changed to $\{N_3\}$ or $\{N_4\}$ since the load of each node is equal. Suppose that $\mathcal{R}(\mathrm{AC}_2)$ is set to $\{N_3\}$. If the load status of each AC holds for a while, the routing information and load status of each node are as follows. □

	N_1	N_2	N_3	N_4
$\mathcal{R}^{-1}$	$\{AC_1\}$	$\{AC_1\}$	$\{AC_2, AC_3\}$	$\{AC_4\}$
#T	75	75	100	50

The notable features of DACA's node expansion strategy are two-fold. First, DACA tries to reduce the number of nodes allocated to the overloaded AC if the load deviation of each node is not significant. This allows DACA to reduce the frequency of buffer invalidations. Next, DACA prohibits allocating both an overloaded AC and other ACs to a node. As a result, DACA can achieve high buffer hit ratio for the overloaded AC. Even though several non-overloaded ACs may be allocated to a single node, we believe that efficient handling of the overloaded AC is more important to improve the overall transaction throughput.

3.3 AC Distribution

AC distribution strategy is applied when a node N_p is overloaded. Among the ACs in $\mathcal{R}^{-1}(N_p)$, suppose that $\#T(AC_{min})$ is the minimum and $\#T(AC_{max})$ is the maximum. If there is a node N_k not allocated to any AC, i.e. $\mathcal{R}^{-1}(N_k)$ is empty, the router updates $\mathcal{R}(AC_{max})$ to $\{N_k\}$. Similarly, if N_k is allocated to an underloaded AC, the router changes $\mathcal{R}(AC_{max})$ to $\{N_k\}$ after performing the node reduction strategy described in the next section. Otherwise, the router changes $\mathcal{R}(AC_{min})$ to include other node, say N_k, where it is the least loaded node in the system.

Example 2: In Example 1, the router allocates both AC_2 and AC_3 to N_3. Now consider the case of increasing the number of active transactions of AC_2. Suppose that the sensitivity factor of α is 1.6 and $\#T(AC_2)$ increases from 50 to 84. The number of transactions in other ACs is assumed to be equal to Example 1. Then $\#T(N_3)$ increases to 134 and N_3 is overloaded since $\bar{L}(N) = \frac{75+75+134+50}{4} = 83.5$ and $134 > 83.5 \times 1.6 = 133.6$. Every node is allocated to some AC and AC_1 is not underloaded. As a result, for AC_2 and AC_3, the router updates $\mathcal{R}(AC_3)$ to N_4 since $\#T(AC_3) < \#T(AC_2)$. □

3.4 Node Reduction

Suppose that AC_q is underloaded and $\mathcal{R}(AC_q)$ is $\{N_q, N_k\}$. If any AC or node is overloaded, the router excludes one of the nodes from $\mathcal{R}(AC_q)$, say N_k, and uses N_k to resolve the overload state. Similar to the node expansion strategy, the node reduction strategy intends to maximize the effect of dynamic load balancing. Transactions of an AC have to be routed to the smaller number of nodes if the routing strategy does not incur any AC overload.

Example 3: After applying the AC distribution of Example 2, suppose that $\#T(AC_1)$ decreases to 50. Then both $\#T(N_1)$ and $\#T(N_2)$ are 25 and $\bar{L}(N)$ becomes $\frac{25+25+84+100}{4} = 58.5$. AC_1 becomes underloaded since $\frac{50}{2-1} = 50 < 58.5$. On the other hand, N_4 is overloaded since $\#T(N_4) = 100 > 58.5 \times 1.6 = 93.6$. The router excludes N_1 or N_2 from $\mathcal{R}(AC_1)$, and then allocates the excluded node to either AC_3 or AC_4 since $\#T(AC_3) = \#T(AC_4)$. □

Table 1. Simulation parameters

System Parameters		
LCPUSpeed	Instruction rate of node CPU	500 MIPS
GCPUSpeed	Instruction rate of GLM CPU	1000 MIPS
NetBandwidth	Network bandwidth	100 Mbps
NumNode	Number of computing nodes	1 – 16
NumDisk	Number of shared disks	10
MinDiskTime	Minimum disk access time	0.01 sec
MaxDiskTime	Maximum disk access time	0.03 sec
PageSize	Size of a page	4096 bytes
RecPerPage	Number of records per page	10
ClusterSize	Number of pages in a cluster	10000
HotSize	Size of hot set in a cluster	2000
DBSize	Number of clusters in database	8
BufSize	Per-node buffer size	4000
MsgInst	Number of instr. per message	22000
LockInst	No. of instr. per lock/unlock pair	2000
PerIOInst	Number of instr. per disk I/O	5000
PerObjInst	Number of instr. for a DB call	15000
LogIOTime	I/O time for writing a log record	0.005 sec
Transaction Parameters		
TrxSize	Number of pages accessed by a trx.	10
SizeDev	Transaction size deviation	0.1
WriteOpPct	Probability of updating record	0.3
MPL	Number of concurrent transactions	10 – 640
ACNum	Number of ACs	1, 8
ACLocality	Probability of accessing local cluster	0.8
HotProb	Probability of accessing hot set	0.8

4 Experiments

In this section, we first describe the simulation model to evaluate the performance of DACA and then analyze the experiment results.

4.1 Simulation Model

We model the SD cluster consisting of a single router and a global lock manager (GLM) plus a varying number of nodes, all of which are connected via a local area network. To compare routing algorithms, the router implements DACA, pure affinity-based routing (*PAR*), and dynamic affinity-based routing that spreads transactions of congested AC to all nodes in a round-robin fashion (*DRR*) [15]. The GLM has a role to perform the concurrency control and the cache coherency control. Two-phase locking is implemented for concurrency control and ARIES/SD [8] for cache coherency control.

Table 1 shows the simulation parameters. Many of their values are adopted from [2,4,15]. We assume that the GLM's CPU performs much better than each node's CPU to prevent the GLM from being the performance bottleneck. The number of shared disks are set to 10, and each disk has a FIFO queue of I/O requests. Disk access time is drawn from a uniform distribution between 10 milliseconds to 30 milliseconds. The *network manager* is implemented as a FIFO server with 100 Mbps bandwidth. The CPU cost to send or to receive a message via the network is modelled as a fixed per-message instruction.

We model that the database is logically partitioned into several *clusters*. Each database cluster has 10000 pages (40 Mbytes), and it is affiliated to a specific AC. The number of ACs is set to either 1 or 8. The transaction parameter of *ACLocality* determines the probability that a transaction operation accesses a data item in its affiliated database cluster. The *HotProb* parameter models "80-20 rule", where 80% of the references to the affiliated database cluster go to the 20% of the database cluster (*HotSize*). We refer the 20% of the database cluster as *hot set*, and the remaining part as *cold set*. The average number of records accessed by a transaction is determined by a uniform distribution between $TrxSize \pm TrxSize \times SizeDev$. The parameter *WriteOpPct* represents the probability of updating a record, and is set to 0.3. The processing associated with each record, *PerObjInst*, is assumed to be 15000 instructions.

The performance metric used in the experiments is a *throughput rate*. The throughput rate is measured as the number of transactions that commit per second. We also use an additional performance metric, *buffer hit ratio*, which gives the probability of finding the requested pages at local or remote buffers.

4.2 Performance of Single AC

To verify the correctness of simulation model, we explore the performance of a single AC by varying *NumNode*, *HotSize*, and *ClusterSize*. Fig. 1 shows the experiment results. The solid lines represent the transaction throughput when *HotSize* and *ClusterSize* are set to the values in Table 1, while *NumNode* is changed from 1 to 16. The transaction router allocates incoming transactions in a round-robin fashion among multiple nodes. The dashed lines show the performance when *HotSize* and *ClusterSize* increase as twice and three times respectively, while *NumNode* is 1. Note that it models the situation where multiple ACs are allocated to a node.

Allocating more nodes to an AC can exploit substantial performance improvement as expected. This is primarily due to the effect of increased buffer hit ratio as Fig. 1(b) shows. When *NumNode* is 1 and *HotSize* is 2000, the buffer hit ratio is around 0.5. Since both *ACLocality* and *HotProb* are 0.8, the probability of accessing hot set is 0.64. Then the buffer hit ratio of 0.5 implies that part of hot set may not be cached in the buffer when *NumNode* is 1. As *NumNode* increases, most pages in the hot set are cached at least one node's buffer, and thus the buffer hit ratio is over 0.64.

An interesting observation is that allocating extremely large number of nodes to an AC shows only marginal performance improvement. In our experiment, allocating over 4 nodes to an AC exhibits similar performance results. If *NumNode* is over 4, most pages in the affiliated database cluster of the AC is cached, but all of database clusters cannot be cached even *NumNode* is 16. As a result, the buffer hit ratios are nearly identical at the area. Furthermore, the potential performance improvement due to large number of CPUs is offset by the increasing probability of buffer invalidation. This motivates our approach of DACA, where DACA allocates more node to an AC when it is *really* beneficial.

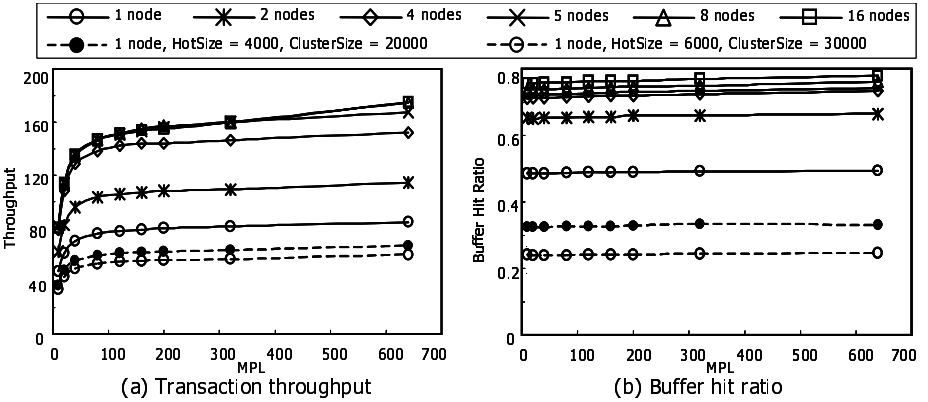


Fig. 1. Performance of single AC

When *HotSize* and *ClusterSize* increase as twice and three times respectively, the performance goes down due to the lower buffer hit ratio. However, their performance differences are relatively small. This is because most of database accesses result in disk I/O.

4.3 Sudden Load Surge

We now compare the performance of three routing algorithms (DACA, PAR, DRR) in case of a load surge of a *single* AC. Fig. 2(a) compares the throughput rate of three routing algorithms. Both *NumNode* and *ACNum* are set to 8. *MPL* is set to 320, and thus the steady state load per each AC before the load surge is 40 transactions. The load surge of an AC is expressed as a fraction of its steady state load. For example, a load surge of 25% implies that the load of each non-surge AC decreases about 25% (10 transactions) and the total sum of additional load (70 transactions) goes to the single surge AC.

When the load serge is 0%, every algorithm performs similarly. As load surge increases, PAR performs worse because PAR does not distribute the extra load of the surge AC to other nodes and thus suffers from lower buffer hit ratio and limited computing facility for surge AC. Fig. 2(b) shows that the buffer hit ratio of PAR is the lowest. When the load surge is 100%, the performance of PAR is nearly half compared to the other two algorithms. In this area, every executing transaction in the system corresponds to the surge AC, and the performance of PAR is equal to that of single AC with 1 node in Fig. 1(a). Both DACA and DRR performs better as the load surge increases. Allocating more nodes to the surge AC can achieve load balancing between nodes. Furthermore, the aggregate buffer space for the surge AC increases also. This is why the buffer hit ratio of both algorithms increases as the load surge increases.

DACA outperforms DRR when the load serge is between 25% and 75%. At first sight, this result might be inconsistent to the buffer hit ratio of Fig. 2(b)

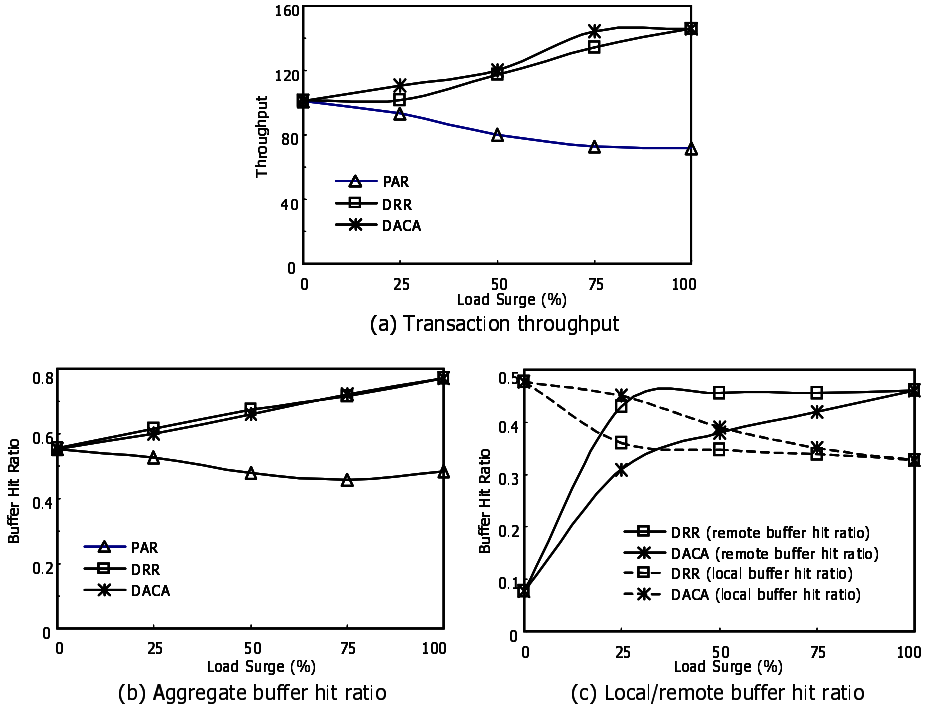


Fig. 2. Performance of sudden load surge

where the buffer hit ratio of DRR is slightly higher than DACA. However, it is important to note that the buffer hit ratio of Fig. 2(b) is the sum of *local* buffer hit ratio and *remote* one. The local buffer hit ratio implies the probability of finding requested data from a node in its local buffer. On the other hand, the probability of finding the data from other node's buffer is represented as the remote buffer hit ratio. Fig. 2(c) shows each ratio separately. Between the load serge of 25% and 75%, the local buffer hit ratio of DACA is higher than that of DRR. The reverse is true on the remote buffer hit ratio. High local buffer hit ratio is more advantageous since the communication overhead due to page transfer between nodes can be reduced. DRR may reduce the amount of disk I/O by distributing the load of surge AC to every node, but it suffers from increasing communication overhead due to high remote buffer hit ratio. Furthermore, the performance improvement of DRR is marginal due to frequent buffer invalidation, which results in low local buffer hit ratio.

5 Concluding Remarks

In this paper, we have proposed a dynamic transaction routing algorithm, named DACA, which selects a node for an incoming transactions so that the load of

each computing node in the SD cluster can be evenly distributed. DACA is novel in the sense that it can achieve an optimal balance between affinity-based routing and dynamic load balancing. We have compared the performance of DACA with other routing algorithms using an SD cluster simulation model. DACA outperforms the pure affinity-based routing algorithm when transaction workload is changed dynamically due to sudden load surge. DACA also outperforms other affinity-based dynamic transaction routing algorithms by judicious node expansion strategy. The previous algorithms suffer from the overhead of frequent buffer invalidation and thus take a lower local buffer hit ratio. On the other hand, DACA can reduce the buffer invalidation effect by allocating more nodes to an AC when it is really beneficial.

References

1. Chen, S., Xiao, L., Zhang, X.: Dynamic Load Sharing with Unknown Memory Demands in Clusters. In: Proc. 21st ICDCS Conf. (2001) 109–118
2. Cho, H.: Cache Coherency and Concurrency Control in a Multisystem Data Sharing Environment. IEICE Trans. on Infor. and Syst. **E82-D** (1999) 1042–1050
3. Cho, H.: Database Recovery using Incomplete Page Versions in a Multisystem Data Sharing Environment. Infor. Processing Letters **83** (2002) 49–55
4. Cho, H., Park, J.: Maintaining Cache Coherency in a Multisystem Data Sharing Environment. J. Syst. Architecture **45** (1998) 285–303
5. DB2 Universal Database for OS/390 and z/OS – Data Sharing: Planning and Administration. IBM SC26-9935-01 (2001)
6. Haldar, S., Subramanian, D.: An Affinity-based Dynamic Load Balancing Protocol for Distributed Transaction Processing Systems. Performance Evaluation **17** (1993) 53–71
7. Hamidzadeh, B., Kit, L., Lilja, D.: Dynamic Task Scheduling using Online Optimization. IEEE Trans. on Parallel and Distributed Syst. **11** (2000) 1151–1163
8. Mohan, C., Narang, I.: Recovery and Coherency Control Protocols for Fast Intersystem Page Transfer and Fine-Granularity Locking in a Shared Disks Transaction Environment. In: Proc. 17th VLDB Conf. (1991) 193–207
9. Nikolaou, C., Marazakis, M., Georgiannakis, G.: Transaction Routing for Distributed OLTP Systems: Survey and Recent Results. Infor. Sciences **97** (1997) 45–82
10. Oracle9*i* Real Application Clusters – Concepts. Oracle Part No. A89867-02 (2001)
11. Shirazi, B., Hurson, A., Kavi, K. (ed.): Scheduling and Load Balancing in Parallel and Distributed Systems. IEEE Computer Society Press (1995)
12. Vallath, M.: Oracle Real Application Clusters. Elsevier Digital Press (2004)
13. Yousif, M.: Shared-Storage Clusters. Cluster Comp. **2** (1999) 249–257
14. Yu, P., Dan, A.: Performance Analysis of Affinity Clustering on Transaction Processing Coupling Architecture. IEEE Trans. on Knowledge and Data Eng. **6** (1994) 764–786
15. Yu, P., Dan, A.: Performance Evaluation of Transaction Processing Coupling Architectures for Handling System Dynamics. IEEE Trans. on Parallel and Distributed Syst. **5** (1994) 139–153

A Personalized Recommendation Agent System for E-mail Document Classification

Ok-Ran Jeong and Dong-Sub Cho

Department of Computer Science,
Ewha Womans University,
Seodaemun-gu, Seoul, 120-750, Korea
{orchung,dscho}@ewha.ac.kr

Abstract. Overload of information due to rapidly developing Internet and increases of e-mails are inconvenience for all Netizens now. Many existing recommendation systems or text classification using personalization techniques are mostly focused on recommending the products for the commercial purposes or web documents. This study aims to apply these application categories to e-mail more necessary to users. Moreover, this study tries to improve the accuracy as eliminating the limits of misclassification that can be key in classifying e-mails by category and deleting Spam mails.
This paper suggests a Personalized Recommendation Agent System (PRAS) recommending the relevant category to enable users directly to manage the optimum classification when new e-mail is received as the effective method for e-mail management. While the existing Bayesian Learning Algorithm mostly uses the fixed threshold, this study proves to improve the satisfaction of users as increasing the accuracy by changing the fixed threshold to the dynamic threshold.

Keywords: Text classification, recommendation agent, e-mail management, Bayesian Learning Algorithm, accuracy, dynamic threshold

1 Introduction

Most recommendation systems recommend the products or other information satisfying the preferences of users on the basis of the users' previous profile information and other information related to product searches and purchase of users visiting websites. This recommendation system is advantageous in dynamically providing the relevant links using the feedback information of other users through the cooperative filtering methods [1,2]. While the application areas of recommendation systems are various such as Usenet news [3], web pages [4,5], video [6], movie, music [7] or books, no researches were conducted for the recommendation systems for e-mail yet. However, the particularity of e-mail needs to be firstly considered to apply these recommendation systems to e-mail. While most information on the webs is for the public, information delivered to e-mails may be considered to be personal in its contents in some degrees. E-Mails consist of the communities where the individual is attached, the companies where the individual joins, the shopping sites where the individual purchased any products or is related to, social acquaintances and unwanted

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 558–565, 2004.
© Springer-Verlag Berlin Heidelberg 2004

mails due to the misuse of an e-mail address. There are e-mail management systems automatically classifying these kinds of mails using the existing text classification. However, the recommendation agent system that can directly reflect the users' opinions and to describe the relevant categories by existing e-mail data will be the most appropriate to flexibly manage the e-mails in accordance with various situations of users.

The E-Mail system has the strong personality so that there will be some problems even if e-mails are automatically classified by category through the learning on the basis of the personal rules. In consideration with this aspect, we need the semi-automatic system enabling both automatic classification and recommendation method to enhance the satisfaction of users. Accordingly, this study uses two approaches as the solution against the misclassification that the users consider as the most serious issue. The first approach is the algorithmic approach to improve the accuracy of classification itself using the dynamic threshold and the second one is the methodological approach using the recommendation agent enabling the users to make the final decision.

The Personalized Recommendation Agent System (PRAS) for e-mail document classification suggested in this paper has the characteristics described below:

- It sets the personal rules as extracting the contents, handling or characteristics of personal mails with respect to the particularity of e-mail. On the basis of setting, this system recommends the relevant category to users in accordance with the priority when a new e-mail arrives.
- This system uses Bayesian Learning algorithm to accurately classify and save mails by category. However, this study improves the existing Bayesian Learning algorithm using the dynamic threshold to reduce the misclassification.
- It provides the interface that can be easily and conveniently used as adding partially automatic and additional functions for the convenience of e-mail users. In addition, it automatically deletes unnecessary or Spam mails.

2 Existing Text Classification Agent System

The prerequisite processes to recommend mails will be basically the text classification. The mails are classified on the basis of the text classification. The mail classification means to allocate each mail to a predefined category. As the number of mails is increased, more times and difficult works are required when effectively searching and indexing mails and summarizing the contents of mails. To solve these problems, it is necessary to classify mails by category and the intelligent machine learning technique using computers instead of heuristic technique. The representative classifications are the nearest neighbor classification, Bayesian Probabilistic classification, decision tree, neural networks, decision rules and support vector machine [8,9]. These classification algorithms have been applied a lot to the feature selection researches for the document classification that is actively researched at present with various methods selecting the features of documents. However, the researches done up to now focused on the effects from selecting syntactic phrases and others when evaluating to decide a specific subset, simply comparing performances by specific selection and selecting features.. This study applies the feature selection

approach dynamically converting the threshold of an existing algorithm by utilizing the particularity of mails rather than how to extract the keyword for the accurate classification.

While numerous researches on e-mail classification systems were conducted including Maxims [10] of MIT that automatically classifies e-mails, the researches on document classification have been actively carried out in various areas. In general, the voluntary software automatically classifying e-mails using the machine learning algorithm is called the classification agent. The representative classification agent is Personal Webwatcher [11] of Carnegie Mellon University. This Personal Webwatcher learns the areas that a user is interested of as monitoring the behaviors of the user through web browsers, classifies what is of interest and what is not of interest with respect to the links in web documents browsed by the user and recommends only the links that the user is interested in. Moreover, InfoFinder[11] developed by the researchers of Anderson Consulting is the agent system searching for the documents what may be of interest to users as classifying online documents on the basis of the profile of user's interest. In addition, Ringo, an entertainment selection agent, and NewT [12], a news article classification agent, are the representative agent systems using the document classification technique.

Furthermore, the successful recommendation systems suing the collaborative filtering are Tapestry [13], GroupLens [14] and PHOKS [5,15].

3 Proposed Application

3.1 Personalized Technique

This study presented idea of using personalized technique for e-mail document classification because there are some problems in applying the existing automatic classifications to e-mails. That is, it is a recommendation agent system for semi-automatic e-mail document classification. E-Mail systems tend to be very personal so that it is difficult to satisfy the users even if the systems automatically classifies e-mails by category on the basis of the personal rules set by learning. As a result, this study recommends the method to classify and automatically delete the mails considered as unwanted mails or Spam and to recommend other mails by category in accordance with the priority when a user reads the mails. The user reading mails by recommended category can prevent the misclassification of mails because they can save the mails in more than one category by priority or manage the mails with personalized techniques when a relevant category is changed due to a time lag. Moreover, the check box is provided in the user interface to make users automatically classify mails when there are too many mails or the reliability on recommendation is satisfied.

3.2 A Personalized Recommendation Agent System (PRAS)

The following Fig. 1 illustrates the filtering, classification and recommendation process by Bayesian learning on the basis of the user's information as the overall layout of PRAS. This system has two characteristics: Firstly, it is configured by

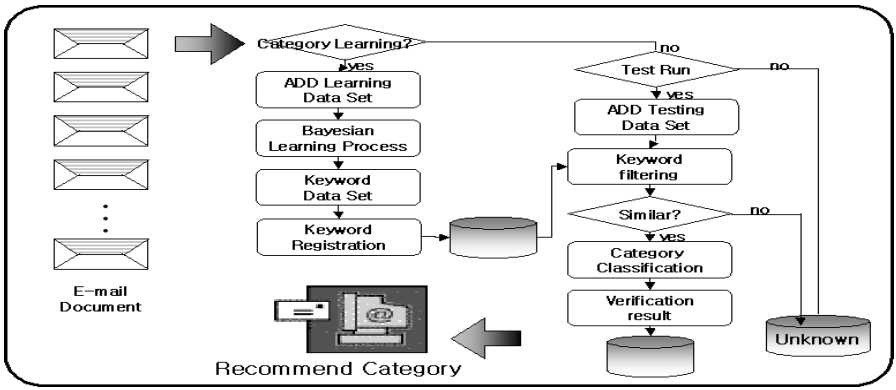


Fig. 1. PRAS Overview

module so that it enables to effectively extract features, generate rules and classify mails by category; secondly, it improves the accuracy of classification as applying Bayesian learning algorithm using the dynamic threshold.

The PRAS consists of three modules as described below. The overall system structure is designed by module and these modules communicate by shared files.

$$Category\ Set\ C = \{c_0, c_1, c_2, c_3, ..., c_k\}, \quad C_0 = unknown\ \ category \tag{1}$$

$$Document\ Set\ \ D = \{d_1, d_2, d_3, ..., d_i\} \tag{2}$$

$$\Re(d_i) = \{p(d_i \mid c_1), p(d_i \mid c_2), p(d_i \mid c_3), ..., P(d_i \mid c_k)\} \tag{3}$$

$$P'_{\max}(d_i) = \max\{p(d_i \mid C_t)\}, \quad t = 1, ..., k \tag{4}$$

$$C_{best}(d_i) = \begin{cases} \{c_j \mid P(d_i \mid c_j)\} = P'_{\max}(d_i), \ \ if\ P_{\max}(d_i) \geq T \\ \qquad where \qquad T = 1 - \dfrac{P_{\max}(d_i)}{\sum_{t=1}^{k} P(d_i \mid C_t)} \\ c_0, \qquad\qquad otherwise \end{cases} \tag{5}$$

The second characteristic is the accuracy of filtering enhanced by dynamically improving the existing fixed threshold. For the formula, we define C in the formula (1) as the entire category group and D in the formula (2) as mail documents. The formula (3) calculates the conditional probability for each category c related to the document D and classifies the document D to the category having the highest probability related to the document D in the formula (4). While the existing Bayesian

learning algorithm fixes the *T* value in the formula (5), this study changes the formula to make the agent dynamically set the *T* value in accordance with the environment learning the mail documents.

3.3 Modular Design

PRAS aims to develop the agents helping uses to handle mails and to provide the user interface facilitating the mail management of users. The overall system is designed by module and each module communicates by shared files. The structure of PRAS by module is illustrated in the Fig. 2 and the detailed roles are explained below.

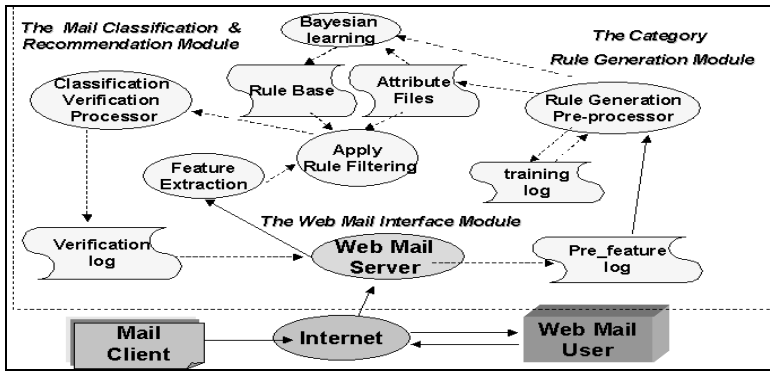


Fig. 2. Modular Design of PRAS

- **The Web Mail Interface Module (WMI):** When a new mail arrives, the system firstly monitors and learns the mail processing of a user. This module helps to extract the features and set the rules. Furthermore, it is the process to set the categories suitable for the personal needs of a user.
- **The Category Rule Generation Module (CRG):** It extracts the features in mail processing and generates the rules in accordance with the personal needs as applying Bayesian algorithm.
- **The Mail Classification & Recommendation Module (MCR):** When a new mail arrives, it classifies mails by category on the basis of defined rules and recommends the mails by category in accordance with the priority. Moreover, it automatically deletes unwanted mails or Spam.

3.4 Implementation of PRAS

PRAS is implemented on the basis of web mails that enable users to log in anywhere anytime, have no limits in a system and requires no mail client programs. It uses Windows 2000 Professional for implementation, MS SQL 2000 for database control and MS visual C++ 6.0, ASP and ASP Components for setting rules and executing algorithm. The user interface is used in the process to monitor users and actually creates or saves categories. A user can create the categories that he/she uses

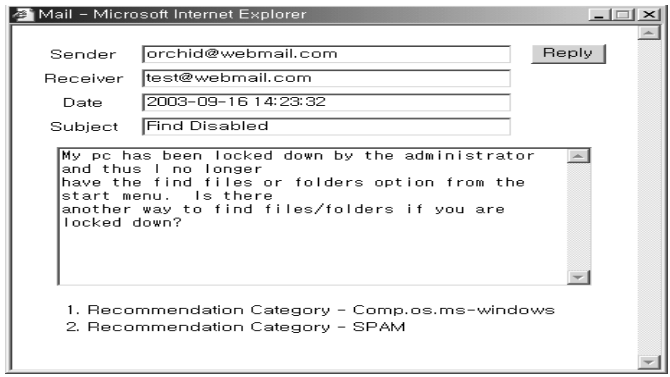


Fig. 3. Recommendation Category

frequently and delete unnecessary categories through the user interface. Moreover, the user interface recommends the categories to a user as internally extracting features and classifying mails during the learning process. When a user reads new mails, he/she receives category recommendation as shown in the Fig. 3.

4 Experiments and Results Analysis

4.1 Precision and Recall

Most of Text Classification (TC) Systems are benchmarked based on their precision and recall. Assume a query is issued to a TC system. Precision is defined as the ration of the number of retrieved documents that are relevant to the user's query over the total number of documents retrieved. Recall is defined as the ratio of the number of relevant documents retrieved over the total number of relevant document known by the system. A perfect TC system would have a precision of 1 and a recall of 1; only relevant documents and all relevant documents would be retrieved. Comparisons between TC systems always hinge on Precision/Recall graphs [3,4].

However, a relevant/irrelevant partition is a dual-class categorization task. This system classifies documents into an arbitrary number of classes, hence the generation of precision /recall graphs would convey considerably less information about the success of the system. A more conventional metric is used instead: classification accuracy. This is valid because the datasets used have similar sizes. It is very difficult to use classification accuracy to provide a fair gauge of success when there are several relevant documents on a database of a few thousand: the two classes are not balanced at all. Missing all of the relevant documents could still provide excellent classification accuracy.

4.2 Classification Accuracy

A user can create the categories that he/she uses frequently and delete unnecessary categories through the user interface. Moreover, the user interface recommends the

categories to a user as internally extracting features and classifying mails during the learning process. When a user reads new mails, he/she receives category recommendation.

The performance evaluation in this study is "how much accurate category the system recommends to the users". This can be demonstrated by checking whether the system properly classifies the mail contents in accordance with relevant categories. For testing the system, we firstly create categories, train the system to learn test data by category and set the rules. Next, we collect the data by category and check whether the system accurately classifies the mails by the predefined rules. For this test, we predefine 12 categories and collect and apply the sample data for the rules and the data for evaluating the performance. The accuracy is checked using FileCheck function among the functions of the system in this test. Since tremendous data needs to be tested, about 10,000 mails by category are tested as one data format.

The results from testing each category in accordance with this method, the results are described in the table 1. In the table 1, the precision 1 is the accuracy when using Beyesian Algorithm and the precision 2 is the result when Beyesian Algorithm using the dynamic threshold is applied.

Table 1. Category Precision Rate (Accuracy)

Number	Category Item	Total data	Correct data(1)	Precision1 (%)	Correct data(2)	Precision2 (%)
1	Comp.graphics	23579	20278	86	21457	91
2	np.os.ms-windo	573	527	92	470	82
3	p.sys.machardv	1578	1404	89	1215	77
4	Misc.forsale	38434	36128	94	38050	99
5	rec.autos	23712	19918	84	20155	85
6	es.sport.baseba	17124	15925	93	16610	97
7	sci.electronics	12354	11242	91	10624	86
8	sci.space	11694	9706	83	9940	85
9	talk.politics.gus	62915	53478	85	59140	94
10	Spam	7260	6461	89	6897	95
11	han.complang	14180	12478	88	12904	91
12	han.plitics	7971	7094	89	7333	92
	total	221374	196137		204796	
	Average			**88.6**		**89.5**

The average accuracy is 88.6%, and 89.5% when the dynamic threshold is applied. The latter accuracy is 0.9% higher than the former one using the existing algorithm. If more time and data for learning is given, the accuracy will be improved more than this result.

5 Conclusion

This study designs and implements a PRAS that can be useful for e-mail users. At present, tremendous amount of information is exchanged through e-mails and the users will request the customized e-mail interface. If more accurate category is recommended as applying the mail documents to the text classifications, the users can

manage their mails more conveniently. Moreover, this study presents two solutions against the misclassification that is the most serious problem in the document classification and the users are worried most. The first solution is the algorithmic approach improving the accuracy of classification itself using the dynamic threshold. The second solution is to use the recommendation that the users can make the final decision without using the automatic classification. When we manage ever-increasing e-mails, the PRAS suggested in this study will be very efficiently used. The future work has to examine the method that the users can directly set the categories. It can improve the classification system as the agent enabling to automatically set and recommend categories at the same time.

References

[1] M. Pazzani, D. Billsus, Learning and Revising User Profiles: The Identification of Interesting Web sites, Machine Learning 27, Kluwer Academic Puglishers, pp. 313–331, 1997.

[2] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based Collaborative Filtering Recommendation Algorithms", Accepted for publication at the WWW10 Conference. May 2001.

[3] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R. and Riedel, J.GroupLens: Applying Collaborative Filtering to Usenet News. *CACM, 40(3)*. 77–87.

[4] Balabanovic, M. and Shoham, Y. Fab: Content-Based, Collaborative Recommendation.*CACM,40(3)*. 66–72.

[5] Hill, W. and Terveen, L., Using Frequency-of-mention in Public Conversations for Social Filtering. *CSCW'96*, 106–112

[6] Hill, W., Stead, L., Rosenstein, M. and Furnas, G., Recommending and Evaluating Choices in a Virtual Community of Use. *CHI'95*, 194–201.

[7] Shardanand, U. and Maes, P., Social Information Filtering: Algorithms for Automating "Word of Mouth". *CHI'95*, 210–217

[8] Ian H. witten and Eibe Frank, Data Mining, Morgan Kaufmann Publishers, Inc., 2000.

[9] Yiming Yang, Jan O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization", *Proc. of ICML97*, pp. 412–420, 1997.

[10] P.Maes, "Agents That Reduce Work and Information Overload", Communications of the ACM, Vol.37, No.7, pp. 30–40, 1994.

[11] Haejung Bak, Yeongdaek Park, Sukhwan Yun, "Web Agent using user's favorite", Korea Information Processing Society Review, September 1999.
http://sslab1.chosun.ac.kr/~chaehwan/study/agent/makeagent_favoriate.htm

[12] Jeffrey M.Bradshaw, "Software agent", AAAI Press/ The MIT Press, pp. 151–161.

[13] Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *CACM, 35*(12). 61–70.

[14] Resnick, p., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J., GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *CSCW'9*4, 175–186

[15] Terveen, L., Hill, W., Amento, B., McDonald, D. and Creter,J. PHOAKS: A System for Sharing Recommendations. *CACM,40*(3). 59–62

[16] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, "Modern Information Retrieval," Addison-Wesley, 1999.

[17] Willian W.Cohen, "Learning Rules that Classify E-Mail", AAAI Spring symposium on Machine Learnning in Information Access, pp. 18–25,1996.

[18] Dunja Mladenic, Marko Grobelnik, " Feature selection for classification based on text hierarchy", *Proc. Of the Workshop on Learning from Text and the Web*, Pittsburgh, USA, 1998.

An Adaptive Prefetching Method for Web Caches

Jaeeun Jeon, Gunhoon Lee, Ki Dong Lee, and Byoungchul Ahn

Yeungnam Unievrsity, School of Electrical Engineering and Computer Science
201-1 Daedong, Gyungsan, Korea
{jenith,gunhoon}@dreamwiz.com, {rhee,ahn}@yu.ac.kr

Abstract. This paper presents a new approach to predict web documents by tracking search patterns of users and by managing documents depending upon the number of hits. Several hot spot documents and their linked documents are stored in cache servers and transmitted them to clients for fast response. The adaptive prefetching method, using search patterns of users, analyzes documents along with the navigation patterns and marks several popular documents. If one of these marked documents is hit, all marked documents are loaded into the cache but only the requested document is transmitted to clients. Cache servers can save their cache memory space as well as provide fast response to clients. The results show that the average response time of the proposed method decreases to 20% compared with other methods and the cache hit rate is increased to 18% of other methods.

1 Introduction

The World Wide Web documents are increasing very rapidly as the internet services are growing very fast. Also the contents of web documents are being changed aperiodically. A lot of web documents including texts and images are transferred through the internet and have important influence on a network performance.

Nowadays, most contents delivered to the networks are multimedia data such as image and digital video. The network traffic generated by multimedia data reduces the bandwidth of networks and gives clients as a slow response. In order to reduce access latency, web documents are stored temporarily near clients and are delivered to clients immediately. The basic principle of web caching is that documents referred once have a very high possibility to be referred again in the near future. Therefore, web caching has become one of interesting research areas to guarantee good performance and fast response.

Web cache servers make decisions based upon document names and URLs. Instead of storing individual document or whole documents of a site, a collection of popular documents accessed several times will save space complexity and time complexity. Also popular documents are likely to be requested repeatedly in the near future. There are some studies about the relation of user access patterns. Tauscher and Greenberg studied that 60 percent of web documents were requested more than once by the same user [10]. In the web client proxy cache study, Duska said that up to 85 percent was the result of multiple users requesting the same documents [11].

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 566–574, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Web caching servers implemented on specialized servers in the network called caching proxies. A proxy requests to a server instead of users and plays the role to return to users with searched documents. Web caching servers can be implemented in various points of the networks. There are three kinds of web caching: forward caching, reverse caching and transparent caching. Forward caching is usually deployed near a user on the edge of a network so that it can serve a many number of users. The use of forward proxy caches benefits in wide-area bandwidth savings with improved response time. Reverse proxy caching is deployed through the network near a server. This is a useful mechanism when supporting web hosting servers. Transparent proxy caches are different from other two methods. This method is not modified by the request of response in anything more than a superficial manner. Then users can not recognize activity of the proxy caches.

Documents which users request repeatedly are saved to a proxy-based caching. Therefore, for the response of users about web document requests, the proxy cache server provides documents to users without going to the original server. As a result, the bandwidth waste and delay of networks are reduced. The possibility to visit identical sites is high when users visit their preferred sites. We propose a variation of the prefetching web caching method to improve the performance and reduce web access latencies by analyzing search patterns of users adaptively.

The following section discusses several prefetching algorithms. In section 3, the proposed algorithm of web prefetching is described. Section 4 describes the experiment results are discussed. And we summarize our observations and conclude the paper in Section 5.

2 Related Work

Two important factors must be considered to deliver web documents to users efficiently. One is the document search of servers and the other is the traffic of networks. The traffic reduction of networks decreases unnecessary bandwidth waste and improves the response time. It is very hard to reduce the load of networks because the connectivity of networks is very complex.

2.1 Web Caching Issues

Many research activities are concentrated on reducing the load of servers by implementing web caching methods. Generally, there are three research areas in web caching: cache replacement policy, cache consistency and prefetching.

Cache replacement policy determines what existing documents in the web cache should be replaced when the web cache is full and new documents are loaded. The goal of replacement policy is to optimize the usage of cache space and to improve cache performance. One complication to implement cache replacement policy in the web cache is that documents to be cached are not the same size as the memory cache. There are some of the important cache replacement policies: first-in/first-out (FIFO), least recently used(LRU), least frequently used(LFU), and so on.

Cache consistency is concerned with ensuring that the documents in the cache are the same as documents in the original web servers. There are four well-known cache

consistency maintenance techniques to deal with detecting such instances: client polling, invalidation callbacks, time to live, and if-modified-since [8]. Commonly If-modified-since is used in the web cache to check cache consistency. Documents are not modified from last access if cache servers receive '304 Not Modified' response.
Prefetching is always an important issue in the web cache. The key idea of web prefetching is to save searching time of documents if they are stored in the web cache. Therefore, some documents are predicted to be used in the near future and stored in the cache. The average waiting time of users can be reduced if documents are hit. The web cache needs to estimate what additional contents will be needed in the next few user access, and store some of them to the local disk or memory. If the prefetched content is indeed requested, the user can access without the delay [1].

Table 1. Comparison of Web prefetching methods. Table 1 shows advantages and disadvantages of four prefetching methods.

Method	Advantage	Disadvantage
Prefetching hyperlinks	Simple computation	Waste of space
Association relation	High hit ratio	Burden of servers
History-based	Low cache space	Difficult to apply in reality
Page rank-based	High hit ratio, Low cache space	Complex computation

2.2 Prefetching Methods

There are four prefetching methods, which are Prefetching hyperlinks, Association relation, History-based prefetching and Page rank-based prefetching. Table 1 shows comparison of four prefetching methods.

Prefetching hyperlinks method uses hyperlinks in documents because it is very high to request a document several times. This prefetching method has applied to all links on the document currently through the web browser [3]. A key drawback of this method is that it typically requires a lot of memory or disk spaces, because all documents of a site must be saved. There is no guarantee that users click the same links on documents. The location and number of documents linked from a document are different. Therefore, if several hundreds links exist on a document, prefetching hyperlinks becomes to be a burden to cache servers without any benefit to users. Finally prefetching weights the load of the networks and brings about the waste of a network bandwidth.

Association relation method maintains a graph each other for document requests of users. It brings all objects to be satisfied specific condition if a user requests one document [4], [5]. This method has had the waste of the bandwidth because it must bring all documents which are related to every request. To maintain the association relation to the graph gives a burden to cache servers.

History-based prefetching uses the history log of servers. Web servers keep the list of popular documents and maintain it by requests of clients or proxies. Proxy brings

the popularity list of web documents periodically from web servers[6]. This method is difficult to apply in reality because all web servers don't have a popularity list.

Page rank-based prefetching approach uses the link structure of a requested document to determine the most important linked document and to identify the documents to be prefetched [10]. If the requested documents have many links to some important documents, those documents have a higher probability of being the next request.

3 Adaptive Prefetching

It is important that cache servers store web documents to be used in the near future. Generally users visit a specific site with similar search patterns. Also web documents of sites are modified or upgraded aperiodically. Therefore, web cache servers update or modify not only documents but also the popularity of documents according to the changes. A new prefetching method is required to improve performance. The underlying premise of the method is that the next documents requested by users are typically based on the previous requested documents. The relative importance of documents is calculated using a hit counter.

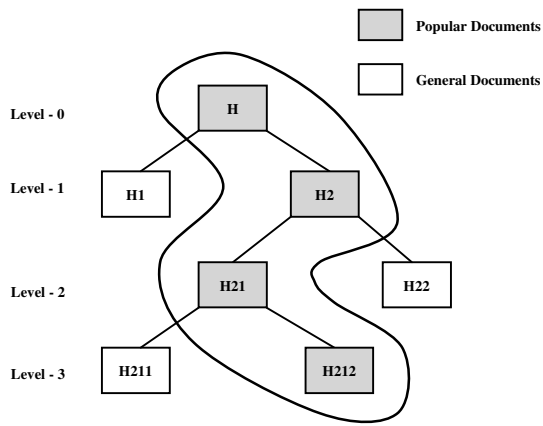


Fig. 1. Search pattern level of cache servers. Level-0 is a site URL and other levels are linked documents searched by users adaptively.

3.1 Adaptive Prefetching Algorithm

When a site is called by users, the main document and the several popular documents are loaded into the cache server at the same time. After documents are loaded, documents are counted by the web cache server whenever they are called.

Fig. 1 shows a tree structure of documents to be searched. Every document of a tree has a hit counter. In Fig. 1, Level-0, root node, means a site home-page. All documents increase their hit counter by whenever they are visited by clients. As time elapses, the shaded nodes have larger hit counters than other documents. If users visit

one of these shaded documents, all shaded documents are loaded into the cache server. In Fig 1, document H, H2, H21, and H212 are loaded into cache.

The adaptive prefetching groups documents and maintain counters as following four steps:

1. Step 1. Set a threshold value for adaptive prefetching. Set all counters to 0. At the initial stage, the cache server operates as the page-ranked prefetching until it collects more than the threshold value.
2. Step 2. If a node is accessed, add all counters of nodes in the path from the root nodes. For example, if node H21 is accessed in Fig. 1, counters of H, H2, and H21 are incremented.
3. Step 3. If a requested document has larger counter value than the threshold, load all documents which are in the link path of the tree. In Fig. 1, if node H212 is requested and its counter value is greater than the threshold value, load H, H2, H21 and H212 into the cache server and increment their counters.
4. Step 4. Update counters of the documents visited by users and decide documents to be removed from the cache or loaded into the cache.

3.2 Adaptive Search Pattern

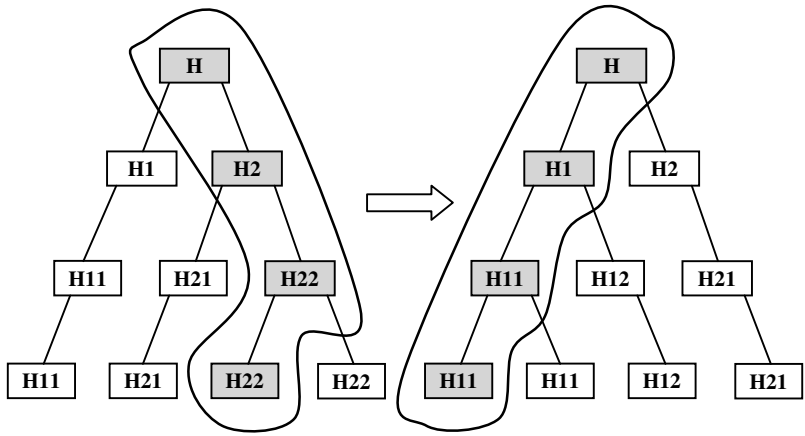
This adaptive prefetching method updates counters of documents to predict the surfing patterns. Fig. 2 shows the adaptive transition of the prefetching document as time elapses. In the beginning, the popular documents are gathered in the right side of the tree. The popularity is changed to documents in the left sided of the tree after time has elapsed.

The cache server uses a threshold value to decide whether documents are loaded into cache. The threshold value is evaluated by the percentage ratio of total visited numbers. The cache server needs to optimize the tradeoff between its resource usage and the hit ratio. The hit ratio indicates how many times requested documents of users hit on the prefetched cache contents. The server resource indicates memory spaces, disk spaces and the cost of processing the prefetcher.

At the beginning, the threshold value is set between 1% to 5% and the cache server will prefetch documents with a relative access probability equal to or greater than the threshold. Since the size of prefetched documents affects the resource usage, the threshold may increase or decrease adaptively. Therefore it prevents the waste of cache spaces and maintains high prediction ratio.

4 Simulation

The squid simulation tool is used to test, verify and evaluate the proposed algorithm[13]. In the simulations, the trace logs of the cache hits are analyzed. Various experiments have been conducted to compare the response times. The results of simulations show that the response time of the proposed method is less than that of the prefetching hyperlinks.



(a) time = 0 (b) time = *n*

Fig. 2. Example of search pattern change. Hot documents in the cache server are changed adaptively as time elapses.

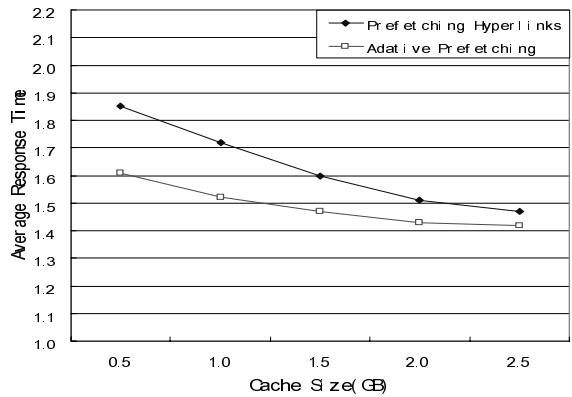


Fig. 3. Comparison of average response time of prefetching hyperlinks and adaptive. The response time of the adaptive prefetching method shows faster than that of the prefetching hyperlinks for smaller cache size.

Fig. 3 shows differences of the response time depending upon the cache sizes. The response time of the adaptive prefetching shows faster response time that of the prefetching hyperlinks for smaller cache size. As the cache size increases, the differences of the response time are smaller and smaller. The reason is that the prefetching hyperlinks method has enough space to hold data in the cache as the cache size is increased. This means that prefetching hyperlinks requires much more cache space for prefetching.

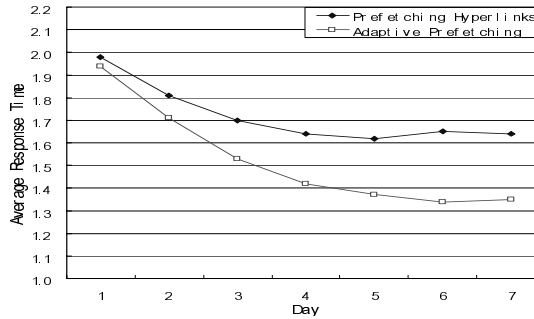


Fig. 4. Comparison of average response time of the prefetching hyperlinks and the adaptive prefetching for a week. The response tome of the adaptive prefetching method shows faster than that of the prefetching hyperlinks as time elapses.

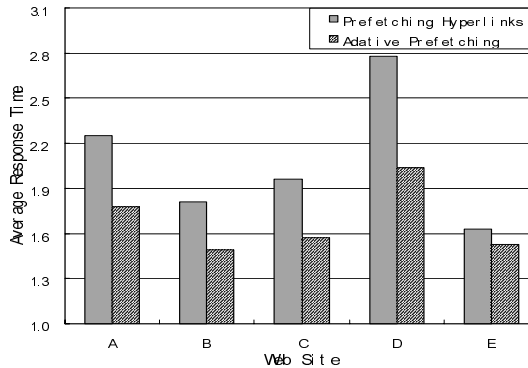


Fig. 5. Comparison of average response time of the prefetching hyperlinks and the adaptive prefetching for the five different web sites.

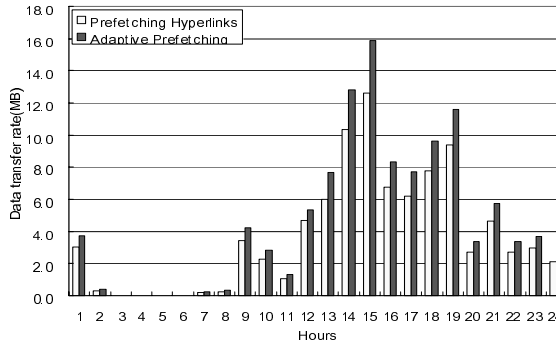


Fig. 6. Comparison of data transfer rate during 24 hours. The adaptive prefetching method transfers more data to clients. This means that cache hit rate of the adaptive prefetching method is higher than that of the prefetching hyperlinks.

Fig. 4 shows that the response time of the adaptive prefetching method shows 20% faster than that of prefetching hyperlinks as the cache server runs several days. At the beginning, the adaptive prefetching method collects data of search patterns. Also it does not have enough information to decide the proper threshold value. Therefore, first several days the response time of the proposed method is very close to that of the prefetching hyperlinks. After several days pass, the adaptive prefetching shows better performance.

Fig. 5 compares the average response time of the two different methods for the five different web sites. The size and depth of the prefetching documents of each site are different. But each site contains documents, graphics, and multimedia data such as audio, video and etc. The results in Fig. 5 show that the adaptive prefetching performs better than the prefetching hyperlinks. In case of the site D, the response time of the adaptive prefetching is about 40 % faster than that of the prefetching hyperlinks.

Data transfer rate is compared to analyze the performance. Data transfer rate during 24 hours is shown in Fig. 6. The adaptive prefetching method transfers data about 18% higher than prefetching hyperlinks does. This means that cache hit rate of the adaptive prefetching method is higher than that of the prefetching hyperlinks. The adaptive prefetching saves and manages more popular documents in the web cache.

5 Conclusion

The proposed algorithm, adaptive prefetching, is tested under the Squid cache environment. The average response time of the adaptive prefetching is much faster than that of the prefetching hyperlinks. The average response time of the adaptive prefetching method is decreased to 20% compared with the prefetching hyperlinks when cache server runs for a week. The adaptive prefetching method shows very good performance for the small cache sizes and provides potential benefits to cache servers which frequent cache replacements are expected. By measuring data transfer rate, the performance of the adaptive prefetching method shows 18% higher. This means that the adaptive prefetching method manages the web cache efficiently.

Since the contents of web sites are modified aperiodically, search patterns are changing continuously. Also the size of documents becomes large and they are stored as disorganized. To solve these problems, the simulation results have revealed the possibility of the adaptive popularity-based prediction method. The adaptive prefetching based upon search patterns of users is an effective web caching method because it satisfies high prediction accuracy as well as a low space requirement.

References

1. Z. Jiang and L. Kleinrock : An Adaptive Network Prefetch Scheme, IEEE J. Selected Areas Comm. Vol. 17, no. 4 (1998) 358–368
2. L. Breslau, P. Cao, L. Fan, G. Philips, and S. Shenker : Web Cache and Zipf-like Distributions: Evidence and Implications, In Preceedings of Infocom (1999)
3. D. Duchamp : Prefetching Hyperlinks : Proc. of Usenix Symp. Internet Technologies and Systems, Usenix (1999) 127–138

4. K. Chinen and S. Yamaguchi : An Interactive Prefetching Proxy Server for Improvement of WWW Latency, In Proceedings of 7th Annual Conference of the Internet Society, Kuala Lumpur, June (1997)
5. Venkata N. Padmanabhan and Jeffrey C. Mogul : Using Predictive Prefetching to Improve World Wide Web Latency, Proceedings of SIGCOMM 96 (1996)
6. E. P. Marcatos and C.E. Chronaki : A Top-10 Approach to Prefetching the Web, In Proceedings of 8th Annual Conference of the Internet Society, Geneva, July (1998)
7. C. Aggarwal and J. L. Wolf and P. S. Yu : Caching on the World Wide Web,IEEE Transactions On Knowledge And Data Engineering (1999) 94–107
8. Barish and K. Obraczka : World Wide Web Caching: Trends and Techniques,IEEE Comm. Magazine (2000) 178–185
9. X. Chen and X. Zhang: A Popularity-Based Prediction Model for Web Prefetching, IEEE Computer Society (2003) 63–70
10. L. Tauscher and S. Greenberg : How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems, Int'l J.Human Computer Studies, vol. 47, no.1 (1997) 97–138
11. B.M. Duska, D. Marwood, and M.J. Feely: The Measured Access Characteristics of World-Wide-Web Client Proxy Caches, Proc. Usenix Symp. Internet Technologies and System (USITS 97), Usenix Assoc., Berkeley, Calif. (1997) 23–36
12. B.D. Davison : A Web Caching Primer," IEEE Internet Computing, (2001) 38–45
13. Squid Web Proxy Cache, http://www.squid-cache.org.

Image Processing and Retinopathy: A Novel Approach to Computer Driven Tracing of Vessel Network

Annamaria Zaia[1], Pierluigi Maponi[2], Maria Marinelli[2], Anna Piantanelli[1], Roberto Giansanti[3], and Roberto Murri[4]

[1] Gerontologic and Geriatric Research Dept., INRCA, Ancona, Italy
a.zaia@inrca.it; annapiantanelli@yahoo.it
[2] Dept. of Mathematics and Informatics, University of Camerino, Camerino, Italy
pierluigi.maponi@unicam.it; meryema@tiscali.it
[3] Geriatric Hospital, Diabetic Unit, INRCA, Ancona, Italy
r.giansanti@inrca.it
[4] Dept. of Physics, University of Camerino, Camerino, Italy
roberto.murri@unicam.it

Abstract. Retinopathy is a major cause of blindness in the world with a higher incidence in senescent people. The most effective treatment to reduce visual loss is early detection of retinal damage through regular screening. We are studying the development of a software for quantifying abnormalities in the retinal vessel network with parameters highly sensitive to pathology progression.
This paper deals with the study of an effective computer driven system of retinal vessel pattern segmentation, able to recognize even thin capillary branching. In particular, the proposed method is mainly based on the so called shape modelling techniques and is tuned up on real retina images. The current version of our method has been tested on several images differing for local retinal vessel pattern view and ocular background. Preliminary results on real retina images are promising and prompt us to follow this approach in building a computer driven segmentation procedure suitable in retina vessel pattern tracing.

1 Introduction

Retinopathy is a major cause of blindness in the world with a higher incidence in senescent people [1]. Basically, three main types of retinal pathology have been identified: diabetic, hypertensive, and senile retinopathy; all of them are characterized by alterations of the vessel pattern [2]. The most effective treatment to reduce partial or total visual loss is early detection of retinal damage through regular screening. Up to date, diagnosis is practically based on qualitative evaluations of retinal morphology, without an objective estimate neither of the degree of degeneration nor of the temporary evolution of the disease. Furthermore, screening of diabetic patients only is a common clinical praxis as they represent a well known high risk population [3]. However, aging of populations is

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 575–584, 2004.
© Springer-Verlag Berlin Heidelberg 2004

a widespread phenomenon in the industrialized world and an increasing incidence of retinopathy has to be expected as diabetes (adult onset type), hypertension, and arteriosclerosis can even co-exist in senescent people.

Digital acquisition systems and image processing techniques allow developing computer driven procedures for detecting and measuring alterations in the retinal images. Using computerized procedures to measure retinal parameters highly sensitive to disease onset and progression would give an effective advantage to both early diagnosis and prevention of the most deleterious complication of vision loss.

These motivations prompted us to face the problem of developing a software for quantifying abnormalities in the retinal vessel network, by using parameters highly sensitive to pathology onset and progression, such to be suitable in both screening and follow-up studies. We have mainly been focusing on two parameters: angles at arteriovenous crossings (AVCAs) and fractal dimension (D_F) of vessel tree, which, in our opinion, can represent good indices of retinal damage evolution. While fractal analysis of retinal vessel tree is under investigation since several years [4], the AVCAs parameter is completely innovative as far as computer driven diagnosis is concerned. Both these parameters are under study in our laboratories and the methods developed for their estimate use different segmentation procedures to be applied. Thus, coming back to the former idea of a software for a quantitative analysis of retinal vessel abnormalities, we focused our attention also on a robust segmentation procedure to be used as the first common step in retina image processing from which applications can start of different methods for several parameters estimate.

The aim of this paper is to describe the approach used to develop our method of segmentation such that it can: i) be applied to retinography images acquired even in the absence of fluorescence dye; ii) represent the starting step for several parameters estimate, from AVCAs at level of large vessels, to D_F of vessel tree at level of even the thinnest capillaries, but also useful for estimating parameters such as vessel tortuosity [5] and vessel diameter [6].

The segmentation of retina images has been considered by several authors and with different methods. As a matter of fact we mention the method of matched filters [7], and the methods based on Mathematical Morphology [8], [9]. The proposed method is given by two different steps: filtering and binarization. The filtering step is mainly based on the method of matched filters. The second one is based on a simple thresholding approach and on the concept of connected components for images.

In section 2, we present the segmentation algorithm and, for convenience of the reader, briefly recall the method of matched filters. In section 3 we report some numerical results. In section 4 some conclusions and further possible developments of the method are discussed.

2 The Segmentation Algorithm

We consider the problem of the segmentation of blood vessel network in retinal images. Let J be a generic gray level image, we suppose that the pixel structure of

J is described by a matrix having real entries. We denote with J also this matrix. Let N, M be the number of columns and the number of rows, respectively, of matrix J; for $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, each entry $J(i, j)$ of the matrix J describes the gray level of the pixel associated to the image J at row i-th and at column j-th.

The segmentation problem can be formulated as follows: let I be a gray level image corresponding to a given retinal vessel network, for each pixel of I determine whether this is or not a vessel pixel. The solution to the problem can be given as a binary image I_s, where the entries of the corresponding matrix are set to one for the pixels belonging to vessels and to zero for the remaining pixels.

In order to perform an accurate segmentation of this kind of images we have to consider the main features of the structure to be recovered, and the properties of the noise usually present in these images. In general, the following properties are assumed for the blood vessel structure:

a) each blood vessel has a piecewise linear pattern;
b) each blood vessel has a Gaussian-like shape along the cross-section direction;
c) the various blood vessels are connected in a tree-like network.

Moreover, the following features are assumed for the noise:

1) low intensity white noise due to the acquisition process;
2) large zones having very different illumination;
3) small zones having non-linear features.

Note that point 1) is a quite common feature in every experimental measurement process. Points 2), 3) describe features peculiar to retinal images. They are not noise really, but add significant difficulty to the segmentation of retinal images; thus, these points must be taken into account for obtaining an accurate result.

We propose a segmentation procedure based on a filtering step for the enhancement of objects characterized by properties a), b), and for the attenuation of objects having properties 1)-3). A binarization step follows by which the blood vessel network is distinguished from the background in the filtered image. The filtering step is mainly based on a simple result of communication system theory. In particular, the filtered version s_T of a signal s is defined as follows:

$$s_T = \mathcal{F}^{-1}(\hat{T}\mathcal{F}(s)), \tag{1}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and the inverse Fourier transform, respectively, T is a transfer function that characterizes the filter, and $\hat{T} = \mathcal{F}(T)$. When s is a disturbed signal, $s = s^* + \epsilon$, where s^* is the exact signal and ϵ is the noise in the signal, the transfer function $\hat{T} = \overline{\mathcal{F}(s^*)}$ gives the best signal/noise ratio. Note that the overbar, in this and other formulas, denotes the complex conjugate. Moreover, among n possible different signals s^*_ν, $\nu = 1, 2, \ldots, n$, a received signal s is recognized as s^*_μ, $\mu \in \{1, 2, \ldots, n\}$ when $\hat{T} = \overline{\mathcal{F}(s^*_\mu)}$ gives the maximum value in (1) with respect to the other possible choices $\hat{T} = \overline{\mathcal{F}(s^*_\nu)}$, $\nu = 1, 2, \ldots, n$, $\nu \neq \mu$, (see [10] pages 307, 416 for details).

This interesting property can be used in the segmentation problem under consideration. More precisely, we have images in place of signals and, according to the previous discussion, we must consider n shape models with properties a), b) in place of n transfer functions. Note that this is a rather usual approach in image processing [7], and in the more general subject of signal processing [10]. A reasonable choice for the shape models is the following one: each shape model is an image where the entries of the associated matrix depend on a given direction, d. They are constant along d and have a Gaussian profile along the direction orthogonal to d. The various shape models differ only for the direction d.

From standard arguments on Fourier transform, we can easily see that formula (1) can be rewritten as a convolution integral involving the signal s and the transfer function T. Let T_ν, $\nu = 1, 2, \ldots, n$, be a given family of shape models. Then, in the segmentation procedure the filtered image I_T is obtained as follows, for $t = L+1, L+2, \ldots, M-L$, $s = L+1, L+2, \ldots, N-L$:

$$I_\nu(t,s) = \sum_{l=-L}^{L} \sum_{k=-L}^{L} T_\nu(l,k) I(t-l, s-k), \ \nu = 1, 2, \ldots, n, \tag{2}$$

$$I_T(t,s) = \max\{I_\nu(t,s), \ \nu = 1, 2, \ldots, n\}. \tag{3}$$

In these formulas it is assumed $L \ll M$, $L \ll N$, and that the relevant information on I is not contained near the boundary of the whole image. In this sense formulas (2), (3) define only the crucial pixels of I_T, the remaining ones being arbitrarily chosen. We note that the filtering procedure given by formulas (2), (3) has a higher computational cost than the corresponding procedure in the Fourier transform conjugate variables, that is, the discrete version of (1). However, procedure (3) is usually more accurate than the procedures yielded from (1).

The shape models are chosen according to properties a), b). In particular, for $\nu = 1, 2, \ldots, n$, we denote with $\theta_\nu = \frac{\pi}{n}(\nu - 1)$ the angle related to the above mentioned direction d, and for $l = -L, 1-L, \ldots, L$, $k = -L, 1-L, \ldots, L$ we define:

$$\tilde{T}_\nu(l,k) = e^{-(l\cos\theta_\nu + k\sin\theta_\nu)/(2\sigma^2)}, \tag{4}$$

$$T_\nu(l,k) = \frac{1}{(2L+1)^2} \sum_{i=-L}^{L} \sum_{j=-L}^{L} \tilde{T}_\nu(i,j) - \tilde{T}_\nu(l,k), \tag{5}$$

where $\sigma > 0$ is a given parameter that controls width and steepness of the profile mentioned in point b). Moreover, parameter L is usually chosen equal to 3σ.

From (5) we have that the mean value of the entries of each T_ν, $\nu = 1, 2, \ldots, n$ is equal to zero. Thus, we can easily see that noises of kind 1), 2) are significantly attenuated by filtering operation (2)-(5). The attenuation is a consequence of the statistical properties for noise of kind 1), while for noise of kind 2) the result follows from the fact that it can be accurately modelled by a locally constant function with respect to the length scale L. Noise of kind 3) is attenuated when the corresponding non-linear zones have size smaller than L. Finally, we note the central role played in the filtering step by the parameter σ that appears in (4). In particular, using a large value for σ, and a corresponding large value for L,

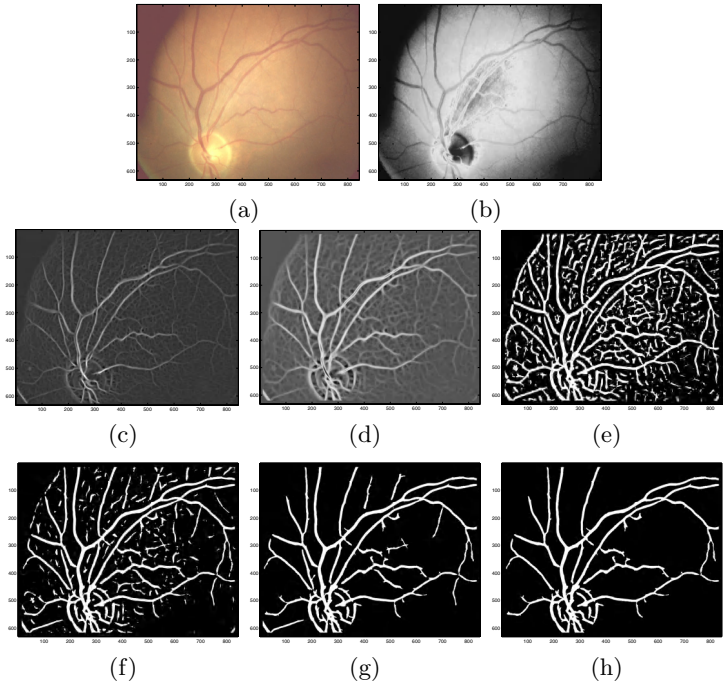


Fig. 1. Results from the steps of segmentation procedure (2)-(8): (a) a color retinal image, (b) the corresponding green channel, (c) the filtered image obtained from (b) using $\sigma = 3$, (d) the filtered image obtained from (b) using $\sigma = 5$, (e) the binary image I_b obtained from (d) with $\tau_1 = 0.01$, (f) the binary image I_b obtained from (d) with $\tau_1 = 0.02$, (g) the segmented image I_s obtained from (f) with $\tau_2 = 500$, (h) the segmented image I_s obtained from (f) with $\tau_2 = 2000$.

we obtain a considerable smoothing effect and a significant enhancement of large vessels while thin ones are attenuated. On the contrary, using a small values of σ and L, we obtain a significant enhancement of thin vessels while large ones can be damaged, and the global noise reduction is usually modest. Examples of filtered images I_T, obtained with two different σ, are shown in Fig. 1(c), (d). Note that σ is large when its value is approximately equal to the half width of large vessels in the considered image and it is small when its value is approximately equal to the half width of the thin capillaries in the considered image.

In the filtering procedure it is implicitly supposed that the gray level of vessels is lower than that of ocular background, that is the case of the RGB channels of retina images (see Fig. 1(b)). The filtering procedure (2)-(5) can deal with gray level images having an opposite situation, i.e. higher gray level of vessels with respect to the background, considering a simple change of sign in formula (3).

From the knowledge of the filtered image I_T we construct a new binary image I_b with the usual thresholding approach. Let $e = \min\{I_T(t,s),\ t = L+1, L+2, \ldots, M-L,\ s = L+1, L+2, \ldots, N-L\}$, $E = \max\{I_T(t,s),\ t =$

$L+1, L+2, \ldots, M-L,\ s = L+1, L+2, \ldots, N-L\}$, let $\tau_1 \in [0,1]$ be a given threshold, then for $t = L+1, L+2, \ldots, M-L,\ s = L+1, L+2, \ldots, N-L$ we define:

$$\hat{I}_T(t,s) = \frac{I_T(t,s) - e}{E - e}, \tag{6}$$

$$I_b(t,s) = \begin{cases} 0, & \hat{I}_T(t,s) \leq \tau_1, \\ 1, & \hat{I}_T(t,s) > \tau_1. \end{cases} \tag{7}$$

Note that normalization step (6) is not essential; however, this is a useful step to obtain a value of the threshold τ_1 independent of the particular retinal image considered (Fig. 1(e), (f)).

The last step in our segmentation procedure is based on the concept of connected components for images (see [11] page 40 for a detailed discussion). Let $C_1, C_2, \ldots, C_p$ be the connected components of subimage $\{(t,s) : I_b(t,s) = 1,\ t = L+1, L+2, \ldots, M-L,\ s = L+1, L+2, \ldots, N-L\}$. Let S be a generic image, we denote with $\#(S)$ the number of pixels of S. Let τ_2 be a positive integer. For $t = L+1, L+2, \ldots, M-L,\ s = L+1, L+2, \ldots, N-L$, the segmented image I_s is defined as follows:

$$I_s(t,s) = \begin{cases} 1, & \exists l \in \{1,2,\ldots,p\} \text{ such that } (t,s) \in C_l, \text{ and } \#(C_l) \geq \tau_2 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

In other words, I_s is given by all the connected components $C_1, C_2, \ldots, C_p$ that have a number of pixels greater than or equal to τ_2. This last step considerably reduces the noise in the segmented image (see Fig. 1(g), (h)).

3 The Numerical Results

We present some numerical results obtained with the segmentation procedure described in the previous section, to provide representative examples of typical retinal images acquired without using fluorescence dye. Retinography color images (632 rows, 843 columns and with 256 color levels in each RGB channel) have been chosen in the database of the Unit of Diabetology, Geriatric Hospital, INRCA. The images taken into account differ for the view of the retinal network; in particular, we consider images with the optic disc (Fig. 2, 3), and images with peripheral retina zones (Fig. 4, 5). Moreover, we also consider images with different backgrounds. Fig. 2(a)-5(a) show examples of retinal images processed to test the efficacy of the proposed segmentation procedure. The numerical results reported below are obtained from the green channel of the considered RGB images (Fig. 2(b)-5(b)). Fig. 2(c)-5(c) show the filtered images. Note that all these filtered images are computed by using formulas (2)-(5) with the following parameters: $n = 16$, $\sigma = 5$, $L = 15$. Finally, from these filtered images and from formulas (6), (7), the binary images I_b are computed choosing $\tau_1 = 0.015$. In (d), (e), (f) of each figure we show the corresponding segmented images I_s computed by using (8) with $\tau_2 = 500, 1000, 2000$, respectively.

The use of real data makes quite difficult a rigorous evaluation of the quality of the results obtained. This difficulty is further increased by the complexity of

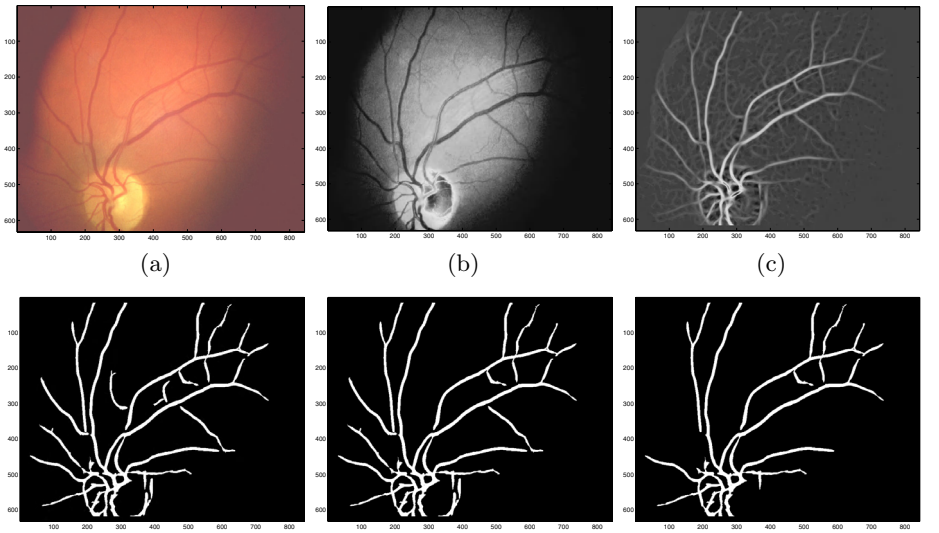


Fig. 2. (a) RGB color retinal image, (b) the corresponding green channel, (c) the filtered image obtained from (b) using $\sigma = 5$, (d), (e), (f) the segmented images I_s obtained from (c) using $\tau_1 = 0.015$, and $\tau_2 = 500, 1000, 2000$, respectively.

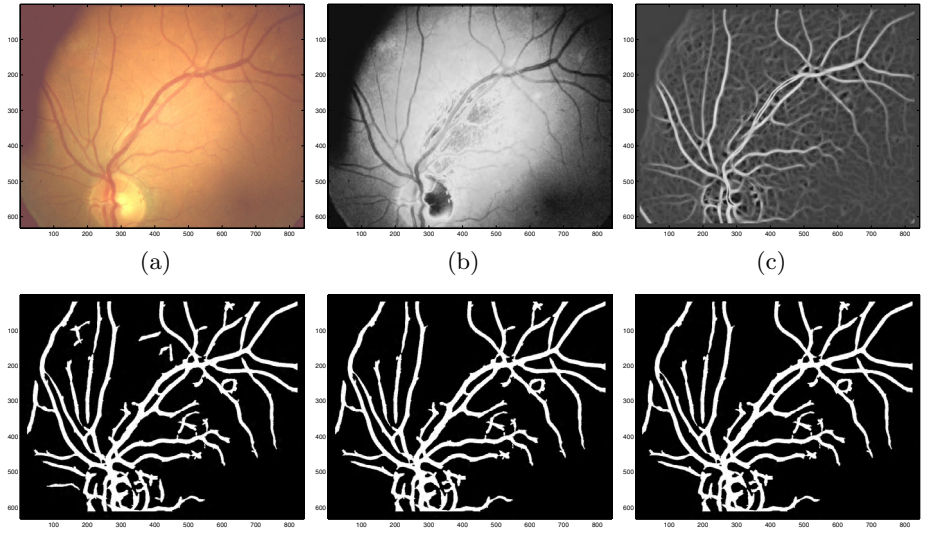


Fig. 3. (a) RGB color retinal image, (b) the corresponding green channel, (c) the filtered image obtained from (b) using $\sigma = 5$, (d), (e), (f) the segmented images I_s obtained from (c) using $\tau_1 = 0.015$, and $\tau_2 = 500, 1000, 2000$, respectively.

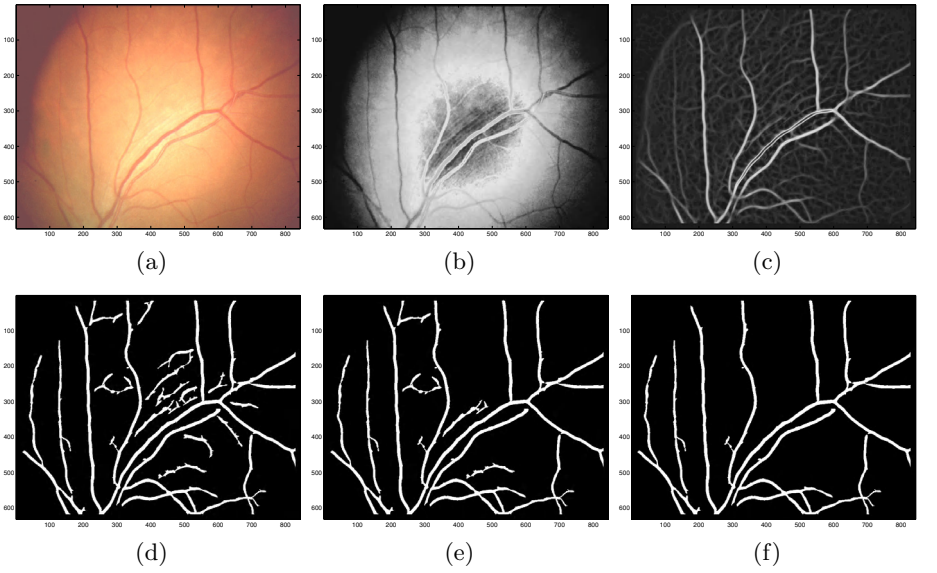


Fig. 4. (a) RGB color retinal image, (b) the corresponding green channel, (c) the filtered image obtained from (b) using $\sigma = 5$, (d), (e), (f) the segmented images I_s obtained from (c) using $\tau_1 = 0.015$, and $\tau_2 = 500, 1000, 2000$, respectively.

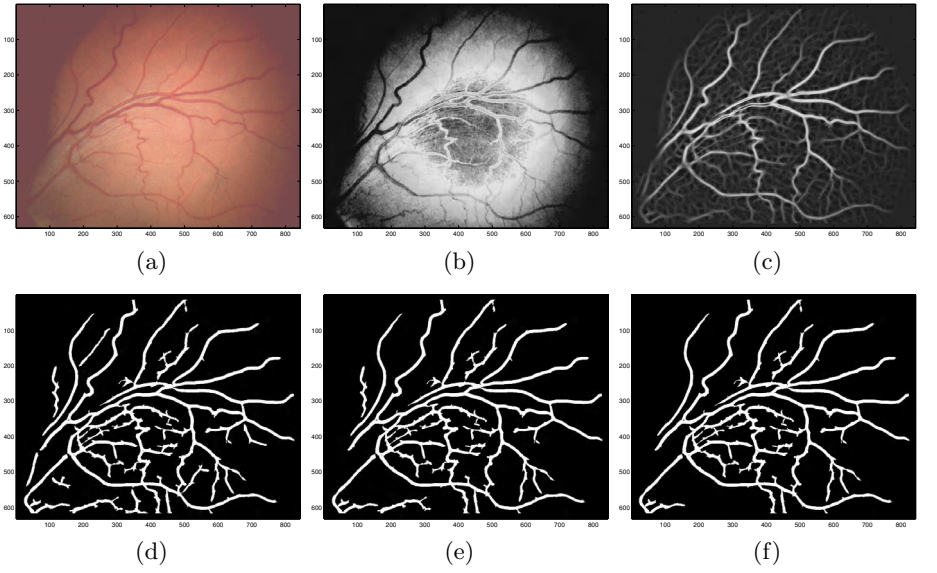


Fig. 5. (a) RGB color retinal image, (b) the corresponding green channel, (c) the filtered image obtained from (b) using $\sigma = 5$, (d), (e), (f) the segmented images I_s obtained from (c) using $\tau_1 = 0.015$, and $\tau_2 = 500, 1000, 2000$, respectively.

the retina vessel network. In scientific reports the presentation of the results is usually based on pictorial arguments comparing original retina images with the corresponding segmented ones. Thus, from Fig. 2-5 the results obtained with the proposed segmentation method can be evaluated as quite satisfactory. These results justify further developments of this version. In particular, the optic disc and the boundary of the objective lens can be source of errors for the segmentation method. However, some simple specializations of the shape modelling technique, to be introduced in future work, allow cutting off the pixels outside the objective lens as well as the pixels belonging to the optic disc, when present. For example, the shape models to recover the optic disc can be based on various circular disk shapes. We finally note that a very promising approach may be given by a clever use of the parameters in the proposed method; for example, the filtered images coming from various choices of parameter σ can be appropriately combined to produce a high quality reconstruction of blood vessels having different width.

4 Conclusions

The proposed segmentation method can be seen as a proper combination of well established image processing techniques, such as shape modelling, thresholding and techniques based on connected components. This method provides quite satisfactory preliminary results on real retina images. In fact, an appropriate choice of the σ value allows tracing the vessel tree in the absence of fluorescence dye. Comparable results have been recently obtained from fluoroangiography images using Mathematical Morphology techniques [8].

Few improvements can be devised. First of all, as above mentioned, using various values for the shape model parameter σ to obtain an accurate segmentation of vascular tree from large vessels to the thinnest capillaries. Moreover, particular attention has to be paid to image fusion techniques [12]. In fact, only the green channel of retina color images is usually processed whereas a fusion of the three channels could provide a higher quality result. For brevity, we cannot deepen this interesting subject; however, evidence of this fact is given by a better definition of the vessel tree in the retina color images (Fig. 1(a)-5(a)) with respect to those in the corresponding green channel (Fig. 1(b)-5(b)).

Such an accurate segmentation procedure could provide advantages to both clinical and experimental research fields. In fact, it can represent the first step to develop a software useful in parametric estimate of retina damages, thus improving both diagnosis and prognosis. In adition, it could allow avoiding fluorescein injection, thus making easier to enlist healthy people in clinical trials. A particular mention deserves aging phenomenon and its interpretation in the light of new paradigms, as it is the theory of complexity [13]. In particular, an innovative hypothesis of work, we are facing on, deals with the application of fractal analysis to gerontology studies. It is based on D_F as an useful tool to measure complexity of biological structures and functions and its modifications with aging and pathology [14]. It could represent a suitable approach to discriminate between physiological and pathological aging as well as between age-related and age-associated diseases, two of the main tasks dealing with aging well.

References

1. Congdon, N.G., Friedman, D.S., Lietman, T.: Important causes of visual impairment in the world today. JAMA **290** (2003) 2057–2060.
2. Vinik, A.I., Vinik, E.: Prevention of the complications of diabetes. Am J Manag Care **9** (2003) S63–80.
3. Bartnik, M., Malmberg, K., Ryden, L.: Recognising and treating the diabetic patient in cardiovascular care. Eur J Cardiovasc Nurs. **1** (2002) 171–181.
4. Landini, G., Misson, G.P., Murray, P.I.: Fractal analisys of the normal human retinal fluorescein angiogram. Curr Eye Res. 12 (1993) 23–27.
5. Hart, W., Goldbaum, M., Cote, B., Kube, P., Nelson, M.: Automated measurement of retinal vascular tortuosity. Proc. AMIA Fall Conference (1997) 459–463.
6. Pedersen, L., Grunkin, M., Ersbll, B., Madsen, K., Larsen, M., Christoffersen, N., Skands, U.: Quantitative measurement of changes in retinal vessel diameter in ocular fundus images. Pattern Recognition Letters **21** (2000) 1215–1223
7. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Trans. Med. Imag. **8** (1989) 263–269.
8. Zana, F., Klein, J.-C.: Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. IEEE Trans. Med. Imag. **10** (2000) 1010–1019.
9. Serra, J.: Image analysis and mathematical morphology. Academic Press, London, 1984.
10. Smith, S.W.: Digital Signal Processing. Elsevier Science, London, 2003.
11. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, Reading, Mass., 1988.
12. Special issue on data fusion. Proceedings of the IEEE. **85** (1997) 1–208.
13. Kyriazis, M.: Practical applications of chaos theory to the modulation of human ageing: nature prefers chaos to regularity. Biogerontology **4** (2003) 75–90.
14. Piantanelli, A., Serresi, S., Ricotti, G., Rossolini, G., Zaia, A., Basso, A., Piantanelli, L.: Color-based method for fractal dimension estimation of pigmented skin lesion contour. Fractals in Biology and Medicine, Vol III, Mathematics and Bioscience in Interaction Series. Eds.: Losa, G.A., Merlini, D., Nonnenmacher, T.F., Weibel, E.R. Birkhauser Press, Basel, 2002, 127–136.

Automatic Extension of Korean Predicate-Based Sub-categorization Dictionary from Sense Tagged Corpora

Kyonam Choo[1], Seokhoon Kang[2], Hongki Min[1], and Yoseop Woo[1]

[1] Dept. of Information and Telecommunication Engineering, University of Incheon
402-749, 177 Dowha-Dong, Nam-Gu, Incheon, Korea
{kyonam,hkmin,yswooo}@incheon.ac.kr
[2] Dept. of Multimedia System Engineering, University of Incheon
402-749, 177 Dowha-Dong, Nam-Gu, Incheon, Korea
hana@tecace.com

Abstract. Analyzing sentences of Korean language, it is found that situation, meaning, and context perform an important role rather than syntactical characteristics. Thus it is difficult to disambiguate word sense by rule-based method, such as context-free grammar, only. In this study, sense-tagged corpora was semi-automatically constructed with the use of predicate-based sub-categorization dictionary. In this process, the information on the frequency of predicate-based sub-categorization patterns, the information on the collocation of predicates and nouns, and the information on the statistic cooccurence of declinable words could be obtained. Based on this information, the method of automatic extension of sub-categorization dictionary is suggested.

1 Introduction

In the situation of a sentence, a head word exists with major meaning. Other lexicons have meaning combinations with the head, and the lexicon that has major role is the predicate, in Korean language. With predicate-based sub-categorization, the constituent dominated by predicate is shown. In case of English, the meaning is determined, with basic background of constituency, by the location of essential constituent in a clause and case markers of prepositions. However, in case of Korean language, the meaning is determined by the relationship with predicate focussed on the case of postposition [1].

The necessity for a sub-categorization dictionary has already been emphasized through many studies [2]. But increased time and effort for construction and increased predicate cause various complicated problems including consistency. Due to these reasons, handing construction was made in very limited areas. There is a limitation in manually constructing, many sub-categorization patterns that are generally utilized. In this study, with utilization of sub-categorization dictionary that had been constructed previously, sense tagged copus were semi-automatically constructed. Based on this work, the method of automatic tuning and extension of sub-categorization dictionary were suggested and identified.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 585–592, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Predicate-Based Sub-categorization Dictionary and Sense Tagged Corpora

2.1 Structure of Predicate-Based Sub-categorization Dictionary

Sub-categorization is one of language information to define the relation of dependency of predicate and complement with clarification of lexical sense of complement. The importance of this information provides with fundamental language resource that might be widely utilized in analysis of sentence structure and meaning. The sub-categorization dictionary utilized in this study was a previously studied one. Not only dependency relationships in sentence structure, which are expressed as case markers on the surface in general, but also semantic roles of complements are attached.

Table 1. Semantic Roles for Predicate-Based Sub-categorization Dictionary

	Semantic Roles	Comment	Postposition of Surface Level
1	AGT	Agent	이/가 [Nominative Case]
2	CHD	Characterized	
3	EXS	Existent	
4	PSS	Possession	
5	TRR	Transforming Result	
6	TRS	Transforming Source	
7	FCS	Focus	
8	CNT	Content	
9	ACC	Accusative	을/를 [Objective Case]
10	PSS	Possession	
11	RNG	Range	
12	TRR	Transforming Result	
13	COM	Commitative	
14	SRC	Source	
15	CNT	Content	
16	COM	Commitative	와/과(Together)
17	CSE	Cause	에/에서/부터 로/으로/으로부터 [Instrumental Case]
18	GOL	Goal	
19	ELM	Element	
20	MEN	Means	
21	INS	Instrument	
22	STS	Status	
23	TRR	Transforming Result	
24	CNT	Content	
25	PTH	Path	
26	RCP	Recipient	
27	MGL	Mental Goal	
28	SCP	Scope	
29	SRC	Source	
30	LOC	Location	
31	CHA	Characterizing	Reserved
32	QNT	Quantity	

Table 2. Case Patterns of Predicate-Based Sub-categorization

	Patterns of Sub-categorization		Patterns of Sub-categorization
V1	[이 AGT]	V25	[이 AGT] [을 ACC] [에/에게 GOL]
V2	[이 AGT] [에서 LOC]	V26	[이 AGT] [을 ACC] [에게(는)/를위하여 RCP]
V3	[이 EXS] [에/에서 LOC]	V27	[이 AGT] [을 ACC] [에(게)서/(으)X로)부터 SRC]
V4	[이 AGT] [에(대하여) MGL]	V28	[이 AGT] [을 ACC] [(때문)에/(으)로 CSE]
V5	[이 AGT] [(때문)에/(으)로 CSE]	V29	[이 AGT] [을 ACC] [을/(으)로 TRR]
V6	[이 CHD] [에 GOL]	V30	[이 AGT] [을 ACC] [(으)로 QNT]
V7	[이 CHD] [에/에서 LOC]	V31	[이 AGT] [을 ACC] [(으)로 STS]
V8	[이 CHD] [(으)로 ELM]	V32	[이 AGT] [에게 RCP] [에(대하여) MGL]
V9	[이 AGT] [에(게)서 SRC] [(으)로(하여) PTH] [에/(으)로 GOL]	V33	[이 AGT] [에게 RCP] [VA고 CNT]
V10	[이 TRS] [이/(으)로 TRR]	V34	[이 AGT] [에게 RCP] [VA다고/라고 CNT]
V11	[이 CHD] [이 CNT]	V35	[이 AGT] [와/에게 ACC] [S기로 CNT]
V12	[이 AGT] [을 ACC]	V36	[이 AGT] [S기로 CNT]
V13	[이 AGT] [을 ACC] [(으)로 INS]	V37	[이 ACC] [VA다고 CNT]
V14	[이 AGT] [을 ACC] [(으)로 MEN]	V38	[이 AGT] [에게 RCP] [VA라고/하라고/게/도록 CNT]
V15	[이 AGT] [(기)를 CNT]	V39	[이 AGT] [에게 RCP] [V자고 CNT]
V16	[이 AGT] [을 CNT] [에 MGL]	V40	[이 AGT] [에게 RCP] [VA느냐고/는가를/는지를 CNT]
V17	[이 AGT] [을 RNG]	V41	[이 AGT] [을 ACC] [에서(부터)/(으X로)부터 SRC] [(으)로 GOL]
V18	[이 AGT] [을 SRC]	V42	[이 CHD] [와 COM] [이 CNT]
V19	[이 AGT] [을 PSS]	V43	[이 CHD] [에게 RCP] [이 CNT]
V20	[이 AGT] [을 COM]	V44	[이 CHD] [로 ELM]
V21	[이 AGT] [와/과 COM]	V45	[이 AGT] [을 ACC] [을 CNT]
V22	[이 CHD] [을 COM]	V46	[이 AGT] [을 ACC] [으로 ELM]
V23	[이 CHD] [와/과 COM]	V47	[이 AGT] [을 ACC] [에대하여 MGL]
V24	[이 AGT] [을 ACC] [와/과 COM]		

And there is an unique character of possible linking with classification system of thesaurus concept. Twenty-five semantic roles are also constructed to express semantic roles of sub-categorization dictionary. These semantic roles are directly related with the cases of surface. Thus, the information of semantic roles directly obtained from syntactical analysis might be utilized in the analysis of meaning structure. Also, a thesaurus of 120,000 lexicons in hierarchical relation is used in expression of concept of noun complements. Forty-seven sub-categorization patterns and seventeen sub-categorization patterns are expressed for verbs and adjectives to express dependency relation of predicates.

Table 3. Example of Predicate-Based Sub-categorization Dictionary

Head Predicate	Information of Essential Complement Elements
가다 (Go)	[Pattern] 1 가(is)[AGT] + 2 를(to)[RNG] [Concept] 1. 인간(Human) 2. 장소(Place)
	[Pattern] 1 가(is)[AGT] + 2 에서(from)[SRC] + 3 로(to)[PTH] + 4 에게(to)[GOL] [Concept] 1. 인간(Human), 인간이외의동물(Animal besides human), 탈것(Things to ride) 4. Place, Human, Direction, Building

With the use of practical scale of thesaurus, a system of substantial selectional restriction for determination of matching of meaning could be constructed through direct application of the thesaurus concept of nouns in sentences to sub-categorization dictionary. Possible inconsistency that might be associated in determination of semantic could be avoided by using standardized predicate pattern based on surface case markers[1,3]. The examples of Korean sub-categorization dictionary are shown in Table 3.

2.2 Construction of Sense Tagged Corpora

Sense tagged corpora that is to be used in obtaining information needed for automatic extension of sub-categorization dictionary was constructed according to the process as shown in Figure 1. Firstly, with use of thesaurus and sub-categorization dictionary, a meaning-tagging module was designed to show automatically the meaning of nouns and dependency relation. A handing correction process was then constructed for sentences with mis-tagging due to algorithmic errors. It was considered that the meaning of nouns and dependency relationships with declinable words are identified through the flow of preprocessor step, subcategory identification step, filtering step, and rank determining step. Through these steps, credible sense tagged copus were constructed for 21,374 sentences.

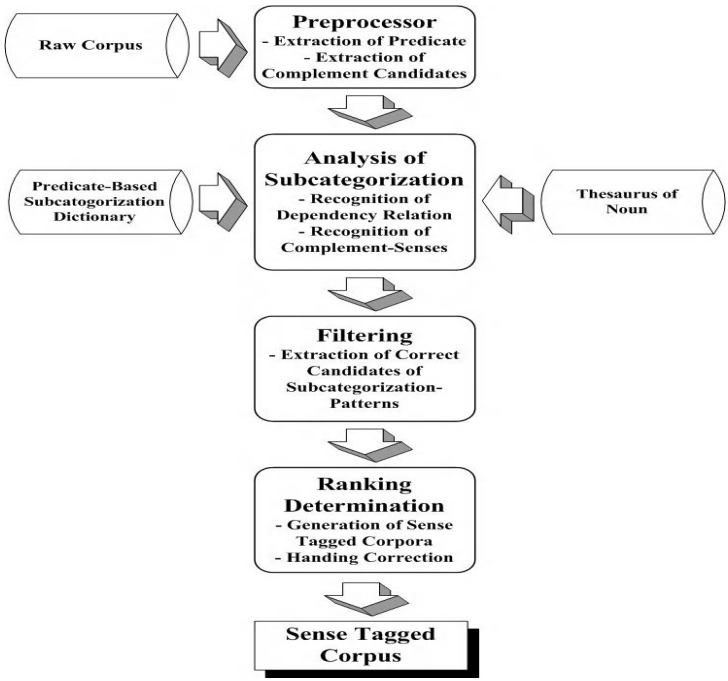


Fig. 1. Process of Sense Tagged Corpora Construction

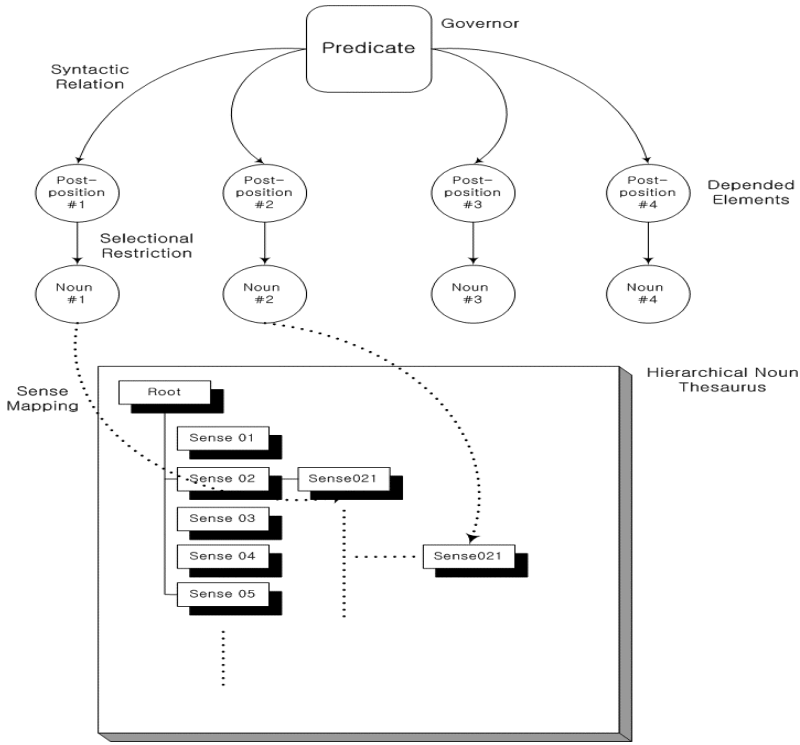


Fig. 2. Structure of Sense-Tagged Corpora

3 Automatic Extension of Sub-categorization Dictionary

In this study, the extension of constructed sub-categorization dictionary was made in two ways. The first was the extension of added meaning, from sense tagged corpora, for each pattern by implementing the frequency and collocation of semantic markers and semantic roles that are the constituent patterns of constructed sub-categorization dictionary. The second was increase of sub-categorization dictionary based on the information post-obtained. Through this process, the accuracy of sub-categorization dictionary could be improved and the rate of matching error could be reduced. Also, failure cases of matching of copus and sub-categorization dictionary were stored separately. Word clustering method was applied to find a hypernym and to add on sub-categorization dictionary.

Addition of contents that are to be added on sub-categorization dictionary was the cases that there is no existence of any of patterns for predicates in current sub-categorization dictionary. The next was the case that, in matching with sub-categorization dictionary, predicates of copus exist in sub-categorization dictionary but the meaning of nouns of essential compliment constituents that are depending on corresponding copus did not match the meaning in sub-categorization dictionary. In this case, it might be happened that patterns match but meanings of nouns do not

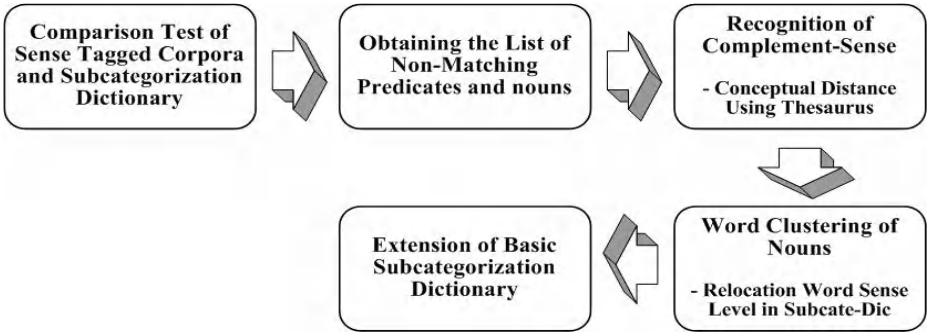


Fig. 3. Process of Automatic Extension of Sub-categorization Dictionary

match. And it might be distinguished into the case of no pattern of matching, again. The process for addition of concept of nouns that did not exist in sub-categorization dictionary was constructed as shown in Figure 3.

4 Experiment and Evaluation

The extension of sub-categorization dictionary was made in two ways. The first was addition of the obtained information obtained from matching test of copus on sub-categorization dictionary. The second was addition of predicates with high frequency on the dictionary. But not all of Korean predicates were included in the dictionary. Thus, the work was addition of predicates and patterns that did not exist in sub-categorization dictionary. The former was automatically included in sub-categorization dictionary through matching test with copus already. It is the case in which predicates of copus do not exist in sub-categorization dictionary. The cases of predicates without appropriate pattern and the cases without matching with meaning were added in sub-categorization dictionary.

The reason that automatically constructed cases of sub-categorization dictionary was not so frequent. The experimental test for extension of sub-categorization dictionary was limited sense tagged corpora that resulted in no performance of automatic extension for predicates associated.

(1) Extension of predicates that does not exist in sub-categorization dictionary

Automatic Extension of Sub-categorization Dictionary	Predicates for Matching	Predicates with No Duplication	Added Patterns
Cases of No Existence in Sub-categorization Dictionary	1492	887	123

(2) Extension of predicates that exists in sub-categorization dictionary

Automatic extension of sub-categorization dictionary	Declinable words for matching	Declinable words with no duplication	Added Patterns
Failure of declinable words with no matching with pattern	798	115	28
Failure of patterns with no concept of nouns matched	13560	3165	845
Failure of patterns with insufficient concept of nouns matched	22037	5165	1213

In this study, to improve accuracy of automatic construction of sub-categorization dictionary, copus tagged with sense were used in semi-automatic method. But the construction of these copus is practically very hard. Thus, mass of corpora should be constructed through utilization of post-constructed sub-categorization dictionary in automatic sense tagging and sentence structure analysis studies. Additionally, repeated performance of extension of sub-categorization dictionary based on corpora is required.

5 Conclusion

The method to add on automatic sub-categorization dictionary was suggested by this study. And the results of automatic extension of sub-categorization dictionary were described through this method. Sense tagging was made automatically in construction of sense tagged corpora in this study to improve the accuracy of automatic extension. Additionally, correction process was performed manually. But handing correction takes lots of time and efforts. The accuracy of automatic sense tagging should be improved to avoid correction process manually. Also, mass of corpora should be implemented in this work.

Sub-categorization dictionary reduces the candidates for sentence structure analysis because it is linked with dictionary of meanings such as thesaurus. Thus mass of corpora might be applied for utilization in construction tagged with information of meanings and for grasp of dependency relation of predicates and complements. Sense tagged copus are automatically constructed with the use of sub-categorization dictionary and thesaurus. Then, extension of sub-categorization dictionary with these constructed copus is performed repeatedly to construct a sub-categorization dictionary containing more information on cooccurence and frequency. In addition, based on this information, a process might be performed to adjust the meaning levels contained in sub-categorization dictionary. With these processes, the construction of more accurate dictionary might be possible.

Acknowledgements. This study was supported by research fund from University of Incheon and Multimedia Research Center of the Korea Science and Engineering Foundation(KOSEF)

References

1. Yoseop Woo, "Constructing a Korean Sub-categorization Dictionary with Semantic Roles using Thesaurus and Predicate Patterns", Paper of Information and Science Association, 2000. 6
2. Atsushi Fujii, “Corpus-Based Word Sense Disambiguation”, Tokyo Institute of Technology , 1998. 7
3. Younghun Seo, "Development of Korean analyzer based on the token-establishment of Korean semantic analysis dictionary and sub-categorization dictionary", Report of Korea Electronic Telecommunication Research Institute, 1998
4. Kyonam Choo, "Korean Lexical Sense Analysis for the Concept-Based Information Retrieval", Paper of master's degree in Incheon university, 1998. 12
5. Yorick Wilks & Mark Stevenson, "Sense Tagging: Semantic Tagging with a Lexicon", Computational Linguistics, 1997. 5
6. Massimo Poesio, "Semantic Ambiguity and Perceived Ambiguity", Computational Linguistics, 1995. 5
7. Ted Briscoe, John Carroll, "Automatic Extraction of Sub-categorization from Copus", Computational Linguistics, 1997. 2

Information Fusion for Probabilistic Reasoning and Its Application to the Medical Decision Support Systems

Michal Wozniak

Chair of Systems and Computer Networks, Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Michal.Wozniak@pwr.wroc.pl

Abstract. Paper deals with the knowledge acquisition process problems. Different experts formulate the rules for decision support system. We assume they have different knowledge about the problem and therefore obtained rules have different qualities. Additional some of them can be generated by machine learning algorithms – of course the quality of information stored in the databases are different. We will formulate the proposition of the confidence measure of knowledge. We will show some of its applications to the decision process. We will propose how use proposed measure for typical probabilistic decision process. The presented concepts will be applied to the real medical decision problem based on boosting idea.

1 Introduction

For decision support systems the quality of the knowledge base plays the key-role. During designing this type of software we get the rules from different sources (experts, databases) and of course their qualities are different. This problem was partly described for the induction learning [1] and for the concept description [2]. In the literature we can find concepts which can be used to the decision making on the base on small learning sets or weak classifiers[3]. Those problems may be interpreted as the decision making on the unreliable (non-representative) sources. The most popular concept (called *boosting*) shows how make the decision using voting procedure.

The very attractive from theoretical and practical point of view is Bayes decision theory[4,5]. The central problem of decision systems implementation of this concept of recognition is how calculate estimator of *posterior* probability. We will propose how to modify boosting methods for probabilistic reasoning. For constructing estimator under consideration we will use weighted sum of weak estimators. Those *posterior* estimators will be obtained on base on rule bases or learning sets. Of course their quality (as the information sources) are different. The value of quality measure (we will propose also).will be use as weights in constructing common estimator.

The following paper concerns on quality of rule for the probabilistic reasoning but the proposed measure can be modified to acquisition process for another form of rule, what we will show in the first part of article. The typical knowledge acquisition process we consider is depicted in Fig.1.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 593–601, 2004.
© Springer-Verlag Berlin Heidelberg 2004

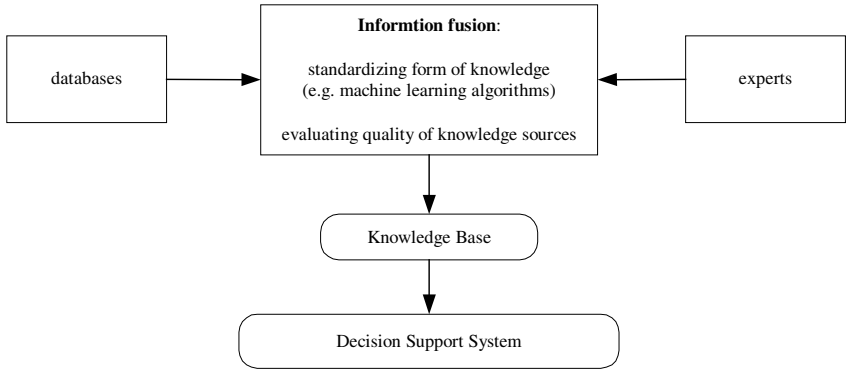


Fig. 1. Idea of the knowledge acquisition process

The content of the work is as follow: Section 2 introduces motivation of the work. Next section presents proposition of statistical knowledge quality measure and the areas of usefulness. Section 4 proposes the rule-based decision algorithm based on the Bayes decision formulae and we present how use proposed quality measure for this method. In section 5 the examination results of proposed decision method for the real medical diagnosis problem is shown. The last section concludes the paper.

2 Motivation of the Work

The main goal of our work is to propose modification of *boosting* method to Bayes decision theory which is used in real medical decision support systems willingly. Boosting is one of the most powerful learning idea[6]. The motivation of *boosting* was a procedure that combines the outputs of many weak classifiers to produce a powerful "committee". A weak classifier is one whose error rate is only slightly better than random guessing. The idea of *boosting* is to produce sequence of weak classifier. The predictions from all of them are then combined through a weighted majority vote to produce the final prediction.

As we see the central problem of proposed concept is how to calculate the weights. This problem is typical for the medical decision support systems, where we can often get only "week" learning materials (rules or learning sets) and finally we can propose "weak" classifiers. Our work presents proposition of problem solution. We propose statistical quality measure of learning materials. It may be calculated as significance level in typical statistical estimation or fixed arbitrary. We propose how it applied to the *boosting algorithm.*

3 Proposition of Knowledge Confidence Measure

3.1 Definition

For typical knowledge acquisition process each of rule is obtained under the following assumption that learning set is noise free (or expert tell us always true). It means

$$P(\text{If A then B}) = 1 . \tag{1}$$

During the expert system designing process the rules are obtained from different sources and each of sources has different quality. For the knowledge given by expert we can not assume that expert tell us true or/and if the rule set is generated by the machine learning algorithms we can not assume the learning set is noise free. Therefore we postulate we do not trust all information we get or we believe on it only with the γ, proposed as the quality (confidence) measure. It can be formulated as [7]

$$P(\text{If A then B}) = \gamma \le 1 . \tag{2}$$

Let $\gamma_i^{(k)}$ denotes the value of proposed measure of rule $r_i^{(k)}$ pointed at the *i*-th class.

3.2 Contradictions Elimination Algorithm

In this section we will show how use proposed quality measure to the typical knowledge acquisition process where if A then B means $A \Rightarrow B$. During this process we may meet with situation that experts formulate the rules whose contradict each other. Lets look on the following example:

Expert 1 said: If A then B
Expert 2 said: If A then C $B \cap C = \varnothing$

In this case, those two rules contradict each other. In this situation we should propose the solution how remove the contradiction.

- Contradiction is detected in rule set given by one expert.
  - Solution: Probably expert wants to cooperate with us to eliminate contradiction from his (or her) set of rules. Expert modifies the conditions of some of rules or remove some of them.
- Rules formulated by different experts are contradict each other.
  - Solution 1: We can ask authors of rules to find the „wrong" rules together and remove or modify some of them.. It is very expensive method, but chance that they want to cooperate is rather small.
  - Solution 2: We decide which of rules should be removed for contradiction elimination (we do not allowed to modify the rule conditions). Probably, we choose rules from the source with the smallest quality (for example from less experienced experts).

For computer implementation of the last solution we can attribute the value of confidence to each of rule.

First we note the set of rule R consists of the M subsets

$$R = R_1 \cup R_2 \cup ... \cup R_M \,, \tag{3}$$

where R_i denote subset of rules pointed at the i-th class.

For this form of rule the two of them contradict each other if

$$\exists\, x \in X \wedge \exists\, k, l \in \{1, 2, ..., M\}, k \neq l \wedge \exists\, i, j \qquad x \in D_i^{(k)} \wedge x \in D_j^{(l)}, \tag{4}$$

where i, j denote the number of rule. The equation (4) means that we can find observation, which belongs to the decision area of the rule pointed at class i ($R_i^{(k)}$) and decision areas of the rule pointed at different class j ($R_j^{(l)}$).

The details of proposition of contradictions elimination algorithm can be found in[9].

4 Probabilistic Decision Support Method

Among the different concepts and methods of using "uncertain" information in pattern recognition, an attractive from the theoretical point of view and efficient approach is through the Bayes decision theory. This approach consists of assumption [8] that the feature vector $x = (x^{(1)}, x^{(2)}, ..., x^{(d)})$ (describing the object being under recognition) and number of class $j \in \{1,2...,M\}$ (the object belonged to) are the realization of the pair of the random variables X, J . For example in medical diagnosis X describes the result of patient examinations and J denotes the patient state. Random variable J is described by the *prior* probability p_j, where

$$p_j = P(J = j)\,. \tag{5}$$

X has probability density function

$$f(X = x \mid J = j) = f_j(x) \tag{6}$$

for each j which is named conditional density function. These parameters can be used to enumerating *posterior* probability according to Bayes formulae:

$$p(j|x) = p_j f_j(x) \Big/ \sum_{k=1}^{M} p_j f_j(x)\,. \tag{7}$$

The formalisation of the recognition in the case under consideration implies the setting of a optimal Bayes decision algorithm $\Psi(x)$, which minimizes probability of misclassification for 0-1 loss function[9]:

$$\Psi(x) = i \;\text{ if }\; p(i|x) = \max_{k \in \{1, ..., M\}} p(k|x)\,. \tag{8}$$

In the real situation *prior* probabilities and the conditional density functions are usually unknown. Furthermore we often have no reason to decide that the *prior* probability is different for each of the decisions. Instead of them we can used the rules and/or the learning set for the constructing decision algorithms.

4.1 Rule-Based Decision Algorithm

Rules as the form of learning information is the most popular model for the logical decision support systems. For systems we consider the rules given by experts have rather the statistical interpretation than logical one.
The form of rule for the probabilistic decision support system[10] is usually as follow

`if A then B with the probability` β,

where β is interpreted as the estimator of the *posterior* probability, given by the following formulae:

$$\beta = P(B|A). \tag{9}$$

More precisely, in the case of the human knowledge acquisition process, experts are not disposed to formulate the exact value of the β, but he (or she) rather prefers to give the interval for its value ($\underline{\beta} \le \beta \le \overline{\beta}$).

The analysis of different practical examples leads to the following form of rule $r_i^{(k)}$:

IF $x \in D_i^{(k)}$ **THEN** `state of object is` i

WITH `posterior probability` $\beta_i^{(k)}$ `greater than` $\underline{\beta}_i^{(k)}$ `and less than` $\overline{\beta}_i^{(k)}$,

$$\beta_j^{(k)} = \int_{D_i^{(k)}} p(i|x)dx\,. \tag{10}$$

For that form of knowledge we can formulate the decision algorithm $\Psi_R(x)$

$$\Psi_R(x) = i \text{ if } \hat{p}(i|x) = \max_k \hat{p}_k(i|x), \tag{11}$$

where $\hat{p}(i|x)$ is the *posterior* probability estimator obtained from the rule set.
The knowledge about probabilities given by expert estimates the average *posterior* probability for the whole decision area. As we see for decision making we are interested in the exact value of the *posterior* probability for given observation. Lets note the rule estimator will be more precise if:

- rule decision region will be smaller,
- differences between upper and lower bound of the probability given by expert will be smaller.

For the logical knowledge representation the rule with the small decision area can be overfitting the training data [11] (especially if the training set is small). For our proposition we respect this danger for the rule set obtained from learning data. For the estimation of the *posterior* probability from rule we assume the constant value of for the rule decision area. Therefore lets propose the relation “more specific” between the probabilistic rules pointed at the same class.

Definition

Rule $r_i^{(k)}$ is "more specific" than rule $r_i^{(l)}$ if

$$\left(\overline{\beta}_i^{(k)} - \underline{\beta}_i^{(k)}\right)\left(\int\limits_{D_i^{(k)}} dx \Big/ \int\limits_{X} dx\right) < \left(\overline{\beta}_i^{(l)} - \underline{\beta}_i^{(l)}\right)\left(\int\limits_{D_i^{(l)}} dx \Big/ \int\limits_{X} dx\right). \tag{12}$$

This definition seems to be very useful for enumerating quality of rules and allows us to formulate the proposition of the *posterior* probability estimator $\hat{p}(i|x)$ is as follow: from subset of rules $R_i(x) = \left\{ r_i^{(k)} : x \in D_i^{(k)} \right\}$ choose the "most specific" rule $r_i^{(m)}$.

$$\hat{p}(i|k) = \left(\overline{\beta}_i^{(m)} - \underline{\beta}_i^{(m)}\right) \Big/ \int\limits_{D_i^{(m)}} dx \,. \tag{13}$$

4.2 Decision Algorithm Based on Learning Set

When only the learning set is given, the obvious and conceptually simple method is to estimate posterior probabilities (7), i.e. discriminate functions of the optimal (Bayes) classifier (8). via estimation of unknown CPDFs (6) and prior probability (5) [4, 5, 12].

4.3 Idea of Recognition Algorithm Based on Boosting Concept

For the considered case, i.e. when *K* knowledge sources are given (sets of rules and learning sets)., we propose the following recognition algorithm:

$$\psi^{(*)}(x) = i \text{ if } \quad p^{(*)}(i|x) = \max_{k \in \mathrm{M}} p^{(*)}(k|x), \tag{14}$$

$$p^{(*)}(i|x) = \frac{\sum\limits_{m=1}^{K} \gamma_m \, \hat{p}^{(m)}(i|x)}{\sum\limits_{m=1}^{K} \gamma_m}, \tag{15}$$

where $\hat{p}^{(m)}(i|x)$ denotes estimator of *posterior* probability obtained on base on the *m-th* knowledge source which value of quality measure is equal γ_m.

5 Diagnosis of Human Acid-Base Balance States

In the course of many pathological states, there occur anomalies in the production and elimination of hydrogen ions and carbon dioxide in the organism, which leads to disorders in the acid-base balance (ABB). We distinguish acidoses and alcaloses, and each of them can be of methabolic or respiratory origin. It leads to the following classification of ABB states: (i) respiratory acidosis, (ii) metabolic acidosis, (iii) respiratory alcalosis, (iv) metabolic alcalosis, or (v) normal state.

Although the set of symptoms necessary to correctly assess the existing ABB disorders is pretty wide, in practice for quick diagnosis, results of gasometric examinations are used, comprising the three following clinical data: (i) pH of blood, (ii) carbon dioxide pressure, (iii) actual bicarbonate concentration.

Acid-base balance disorders have a dynamic character, quickly change in time depending on the previous states and the applied treatment, and required frequent repetition of examinations so that the up to date state of the ABB could be determined. All this makes the above presented recognition methodology appropriate for computer-aided recognition of ABB states.

The presented model should be enhanced with the therapeutical (control) procedures that the patient is subjected to after every examination. These therapies can be comprised in the following three categories: (i) respiratory treatment, (ii) pharmacological treatment, (iii) no therapy.

In the Neurosurgery Clinic of Wroclaw Medical Academy the set of data has been gathered, which contains 88 learning sequences (each sequence consists of 10 to 25 observations) and the set of 55 rules in the form presented in .point 4.1.

In order to study the performance of the proposed recognition concept and evaluate its usefulness to the computer-aided diagnosis of human acid-base balance states some computer experiments were made.

For computer simulation experiment we choose four classifier

1. k(N)-NN algorithm (curve k-NN),
2. rule-based algorithm where each of rule has the same confidence measure value (curve GAP),
3. algorithm based on boosting. We decide that the confidence measure value of each type of information is the same (learning set and rules). For this classifier mixing factor value was fixed on ½ (curve komb const),
4. combined classifier which mixing factor value depends of confidence value of information (curve komb var). For calculate the mixing factor value α we used the following heuristic formulae:
   $\alpha(x_n) = 0$ if no rule is active for given observation x_n,
   $\alpha(x_n) = 10/(10+m)$ if any of rule is active for given observation x_n, where m is the number of learning sequence.

   The results of test are presented in Fig. 2.
   On the base on the experimental results we can make the following conclusion:

- Combined algorithms lead to better or similar results compared to algorithm k-NN, especially for the big learning set.
- Combined algorithm with mixing factor value depends of confidence value of information leads to the higher frequency of correct classification than combined one with constant value of mixing factor.

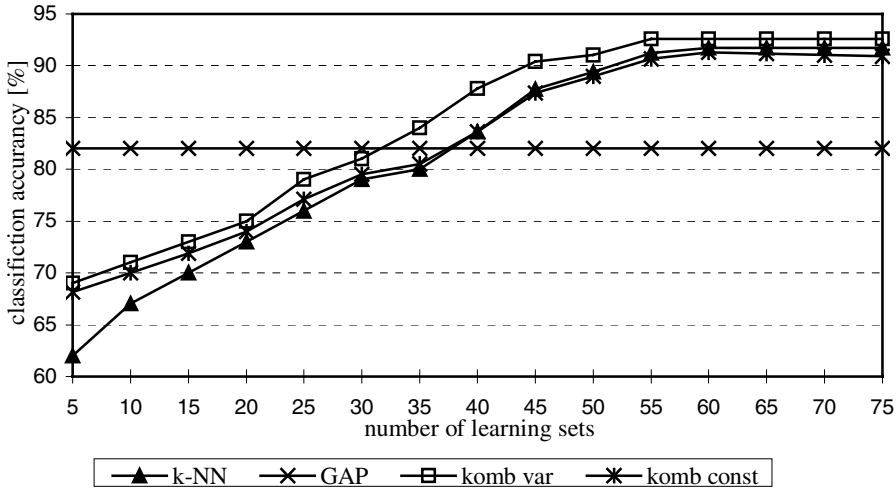


Fig. 2. Results of classification accuracy [%] versus the number of learning sets for GAP, k-NN, combined with constant and mixed factor value algorithms

It must be emphasized that we have not proposed a method of "computer diagnosis". What we have proposed is the algorithm which can be used to help the clinician to make his own diagnosis. The superiority of the presented empirical results for the combined algorithm over GAP and *k*-NN algorithms demonstrates the effectiveness of the proposed concept in such computer-aided medical diagnosis problems in which both the learning set and expert rules are available.

6 Conclusion

The paper concerned probabilistic reasoning and the proposition of the quality measure for that formulated decision problems. We proposed the idea of the contradiction detection and elimination method for the logical representation of experts' knowledge too. We hope this idea of confidence management can be helpful for other problems whose can be met during the knowledge acquisition process from different sources.

The central problem of our proposition is how to calculate the confidence measure. For human experts the values for their rules is fixed arbitrary according to the quality of creator. The presented problem we can also find in the typical statistical estimation of unknown parameter β, where we assume the significant level. The significant level can be interpreted as the confidence measure.

Presented ideas need the analytical and simulation researches but the preliminary results of the experimental investigations are very promising. Let us draw some future works under the concept of the information quality:

1. developing the method how to judge the expert quality (we formulated only the method of counting the confidence measure for rules obtained via machine learning algorithms but we propose arbitrary judgment for rule given by experts),

2. analytical researches into proposed method properties,
3. performing simulation experiments on computer generated data to estimate the dependencies between the size of the decision area and the data quality versus correctness of classification.

Acknowledgement. The work presented in this a pat of the project *Pattern recognition methods based on boosting concept – comparative study* supported by The University of Applied Sciences in Legnica.

References

1. Dean P., Famili A., Comparative Performance of Rule Quality Measures in an Inductive Systems, *Applied Intelligence*, no 7, 1997.
2. Bergadano F., Matwin S., Michalski R.S., Zhang J., Measuring of Quality Concept Descriptions, *Proc. Of the 3rd European Working Session on Learning*, Aberdeen, Scotland, 1988.
3. R. E. Schapire. The boosting approach to machine learning: An overview. Proc. Of MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, 2001.
4. Devijver P. A., Kittler J., *Pattern Recognition: A Statistical Approach*, Prentice Hall, London 1982.
5. Puchala E., A Bayes Algorithm for Multitask Pattern Recognition Problem – Direct Approach, Springer Verlag Lecture Notes in Computer Science, no 2659, 2003.
6. Hastie T., Tibshirani R., Friedman J., The Elements of Statistical Learning; Data Mining, Inference*, and Prediction*, Springer-Verlag, New York 2001.
7. Wozniak M., Application of the Confidence Measure in Knowledge Acquisition Process, Springer Verlag Lecture Notes in Computer Science, no 2657, 2003.
8. Duda R.O., Hart P.E., Stork D.G., *Pattern Classification*, John Wiley and Sons, 2001
9. Puchala E., The complex pattern recognition algorithms - comparative analysis, [in:]. Proc of the 37th Conference Modeling and Simulation of Systems, Ostrava, Czech Republic 2003.
10. Giakoumakis E., Papakonstantiou G., Skordalakis E., Rule-based systems and pattern recognition, *Pattern Recognition Letters*, No 5, 1987.
11. Mitchell T., Machine Learning, McGraw Hill, 1997.
12. Kasprzak A., Exact and Approximate Algorithms for Topological Design of Wide Area Networks with Non-simultaneous Single Commodity Flows, Springer Verlag Lecture Notes in Computer Science, no 2660, 2003.

Robust Contrast Enhancement for Microcalcification in Mammography

Ho-Kyung Kang[1], Nguyen N. Thanh[1], Sung-Min Kim[2], and Yong Man Ro[1]

[1] Multimedia group, Information and Communication University, Korea
{kyoung,nnthanh,yro}@icu.ac.kr
[2] Department of Biomedical Engineering, School of Medicine, KonKuk University, Korea
smkim@kku.ac.kr

Abstract. Microcalcification is important for early breast cancer detection. But due to the low contrast of microcalcifications and same properties as noise, it is difficult to detect microcalcification. In this paper, we propose a robust contrast enhancement method for microcalcification. The proposed method is modified homomorphic filtering in wavelet domain based on background noise information. By using the proposed method, the mammogram contrast can be stretched adaptively thereby enhancing the contrast. Experimental results show that the proposed method improves the visibility of microcalcifications. The contrast improvement index (CII) is increased while noise standard deviation is decreased.

1 Introduction

Breast cancer is currently the leading causes of death among middle-aged women. In the medical perspective, the earliest symptom of breast cancer is the appearance of microcalcifications. Thus, the detection of microcalcification is a major part of diagnosis in early stage breast cancer. However, microcalcification is too small to detect by palpable breast diagnosis. Mammography is known as the best modality to detect microcalcification [1].

The small size of microcalcification results in poor visualization in mammograms. Therefore, to provide the improved visibility of breast cancer to medical doctors as well as automatic breast-cancer detection systems, mammogram contrast should be enhanced. In doing so, denoising is considerable for image enhancement. Especially for mammogram, the size of microcalcification is close to noise's. Noise should be reduced while microcalcifications are enhanced.

Some enhancement methods for mammogram have been proposed [2, 3]. However, in these methods, noise properties of mammogram were not considered properly. Some mammograms are taken from different environments such as different noise condition, X-ray intensity and concentration of sensitizer of mammogram films. Mammography in different noise conditions should be considered.

In this paper, we propose a robust image enhancement and noise reduction by using noise characteristics in background region of each mammogram. This paper consists of the flowing sections. Section 2 presents basic theory of contrast enhancement. In section 3, we describe the robust contrast enhancement method. Section 4 presents the experimental results. We make conclusions in section 5.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 602–610, 2004.
© Springer-Verlag Berlin Heidelberg 2004

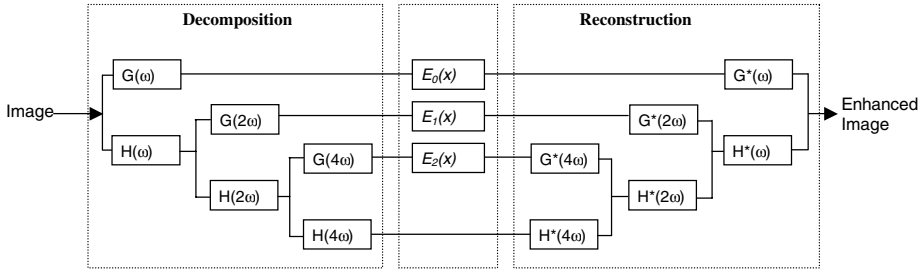


Fig. 1. One dimensional contrast enhancement in wavelet domain

2 Contrast Enhancement in Wavelet Domain

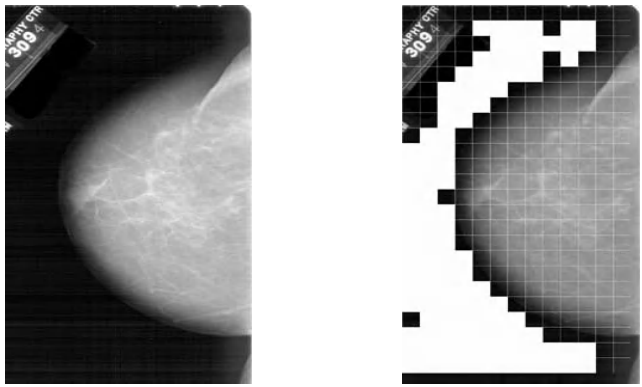
The basic idea of wavelet transform is to analyze different frequencies of the signal using different scales. High frequencies of the signal are analyzed using low scales and low frequencies are analyzed in high scales. This is a far more flexible approach than the Fourier transform, enabling analysis of both local and global features [11]. The 1-D discrete dyadic wavelet transform is shown in Fig. 1. Left box shows the part of signal decomposition while right box is signal reconstruction part [2, 3].

In Fig. 1, $G(\omega)$ is a high pass filter and $H(\omega)$ denotes a low pass filter. $G^*(\omega)$ is a reconstruction filter of high pass components and $H^*(\omega)$ is a reconstruction filter of low pass components. The variable of 2ω and 4ω indicates a sub-sampling of input signal. 1-D signal can be decomposed into three high pass channels ($G(\omega)$, $G(2\omega)$ and $G(4\omega)$) and one low pass channel ($H(4\omega)$). $E_0(x)$, $E_1(x)$ and $E_2(x)$ are gains of each wavelet channel.

3 Robust Image Enhancement in Wavelet Domain

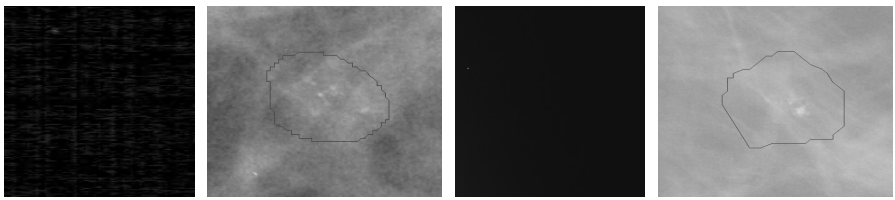
A mammogram is divided into three distinctive regions: the breast region, the background (non-breast) region, and the regions of artifacts. The breast region is created when X-ray is absorbed in breast. Background is the region where X-ray has no obstacle. Artifacts are objects such as labels. Background segmentation is useful for computer-aid-system because it significantly reduces the checking area. Moreover, the background region gives us information of noise that is used for processing the breast region.

In the previous contrast enhancement methods, the parameters such as gains of filtering and thresholds of denoising are usually the fixed value that is same for all mammograms. The propose method, in this paper, noise characteristics of each mammogram is considered in homomorphic filtering as well as denoising process. Properties of noise are obtained from the background region.



(a) Original image (b) Background segmentation

Fig. 2. Result of background segmentation



(a) (b) (c) (d)

Fig. 3. Examples of background noise and microcalcification areas (indicated by white arrows). (a) and (b) are high noise (var>40) in background and breast, (c) and (d) are low noise (var<20) in background and breast area (12 bit gray level)

3.1 Background Extraction in Mammogram

In the proposed method, background segmentation is performed based on the properties of mean and variance of gray level.

Fig. 2 shows an example of block segmentation. (a) is the original image, and (b) is the result of background segmentation.

Mammograms in DDSM (Digital Database for Screening Mammography) include various kinds of noise. In some images, the noise is diffused through background and breast area while others are very clean. Fig. 3 gives two examples, in which (a) and (b) are two parts of a mammogram with high noise condition and (c) and (d) are low noise case. In Fig. 3 (b), we can notify some noise characteristics which are similar to Fig. 3 (a).

3.2 Robust Mammogram Enhancement Using Homomorphic Filtering

The homomorphic filter function decreases the energy of low frequencies while increases those of high frequencies in the image. The homomorphic filter is used to

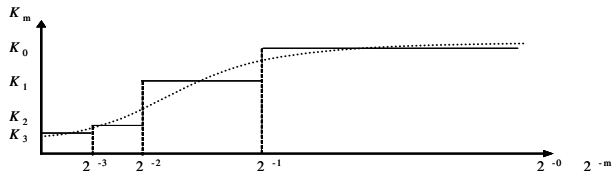


Fig. 4. Homomorphic filter function for applying to wavelet coefficients

find the gain K_m [3]. With the mammogram, the homomorphic filter gives contrast stretching for lower gray level by compressing dynamic range of the gray level. Based on the characteristics of homomorphic filter function, we determined the gain of mapping function, *i.e.*, weighting wavelet coefficients of channels corresponding to homomorphic filter function. Fig. 4 represents the gain K_m that is determined according to the discrete homomorphic filtering. Note that the dotted line in Fig. 4 represents continuous homomorphic filter function.

In mammogram contrast enhancement, noise reduction is a considerable issue. One method of denoising is wavelet shrinkage that was presented in [3]. Each mammogram contains its own noise characteristics because mammograms are taken from different environments as shown in Fig. 3. Therefore, applying the same parameters in noise reduction and the gain K_m for every mammogram are not efficient. Taking into account noise properties of each mammogram, we propose robust method for mammogram enhancement.

To obtain noise characteristics of the mammogram, the background is segmented by thresholding the values combining the gray-level, mean and variance of pixels. The segmented background areas are supposed to contain the noise of image. Therefore, we can take noise characteristics in this area. The noise characteristic is measured by background noise variance (var_b), which can be written as

$$\mathrm{var}_b = \frac{1}{N_b}\sum_{(x,y)\in background}\{(I(x,y)-mean(x,y))^2\}, \tag{1}$$

where N_b is the number of pixel in background area. $I(x,y)$ and $mean(x,y)$ are calculated using background pixels only. If high variance of background noise exists, we need to reduce the gain of homomorphic filter in high frequency domain as

$$K_m' = K_m \times \frac{A}{\mathrm{var}_b + A}, \quad if\ m = 0\ and\ m = 1 \tag{2}$$

where A is constant value to normalize noise variance, m is a level of wavelet and K_m means the gain of each wavelet level. In equation (2), m=0 means highest frequency level in wavelet and m=1 means second highest wavelet level. In high noise mammograms, the gains are reduced whereas, in low noise mammograms, higher gain of contrast enhancement is acceptable.

Fig. 5 is a diagram of modified homomorphic filtering approach in the proposed framework. Fig. 5 shows 3 level wavelet decomposition and reconstruction with one-dimensional signal. Here, we first take logarithmic function for input signal. It also inverts the exponential operation caused by the radioactive absorption, which is generated in the process of obtaining mammography image. K_m` is linear enhancement gain of each wavelet channel. It is suited for enhancement of microcalcification because it emphasizes strong edge much more than the weaker edge [2].

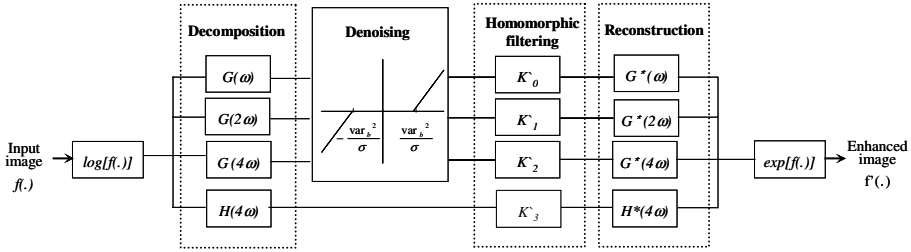


Fig. 5. Robust contrast enhancement with denoising and modified homomorphic filtering

Further, an adaptive denoising is included in the enhancement framework in wavelet domain shown in denoising block of Fig. 5. To achieve the edge-preserved denoising, a nonlinear wavelet shrinkage method is applied. In denoising, wavelet coefficient values are reduced to zero according to a level-dependent threshold. Noise adaptive shrinking operator ($S(u)$) for the denoising can be written as

$$S(u) = \begin{cases} sign(u) \times \left(|u| - \mathrm{var}_b^{\,2} / \sigma \right) & if\ |u| > \mathrm{var}_b^{\,2} / \sigma \\ 0 & otherwise, \end{cases} \tag{3}$$

where u is wavelet coefficient, and σ is variance of reconstructed image using wavelet coefficients in a sub-band. *sign(u)* means positive or negative sign of u. Threshold in this wavelet shrinkage is called a nearly optimal threshold [7].

Taking modified homomorphic filter gains of high frequency area in wavelet domain and optimal denoising operators, microcalcification can be enhanced and also noise can be reduced in breast area.

4 Experiment

To verify the proposed method, experiments are performed with DDSM mammogram database. In DDSM database, the resolution of a mammogram is 50 μm/pixel and gray level depths are 12 bits and 16 bits with various kinds of noise characteristics.

In the experiment, three contrast enhancement methods are performed: linear enhancement in wavelet domain (unsharp marking) [2], homomorphic filter in wavelet domain (homomorphic filtering) [3], and the proposed robust enhancement using modified homomorphic filter in wavelet domain (proposed enhancement). Because the proposed method is to enhance contrast of microcalcification and reduce noise, parameters of enhancement methods were chosen suitably for this purpose. If thresholds of denoising are small, much noise will remain after denoising and this noise is enhanced by filtering. If the thresholds are so high, mammogram will be burred and small microcalcifications will be eliminated.

A quantitative measure of contrast improvement is calculated using contrast improvement index (CII) [2]. CII is defined as

$$CII = \frac{C_{enhanced}}{C_{original}}, \tag{4}$$

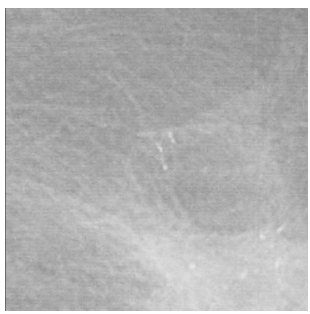
where $C_{enhanced}$ and $C_{original}$ denote for the contrast values of microcalcifications in the enhanced and the original images, respectively. The contrast C of a microcalcification in the image is defined as

$$C = \frac{f - b}{f + b}, \tag{5}$$

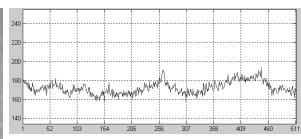
where f is the mean value of the microcalcification, and b is the mean value of background. The standard deviation (std.) of pixels in background region is also measured in order to represent the level of noise.

Fig. 6 shows enhancement results for a high noise image. The profiles ((e), (f), (g) and (h)) contain microcalcifications in center. The contrast improvements of both homomorphic filtering and proposed enhancement methods are equivalent with center peak of profile in Fig. 6, and CII value of Table 1. However, Fig. 6 (g) and (h) show that the proposed enhancement much better than the homomorphic filtering in denoising. This is also indicated by the standard deviation of noise (std. of noise) in Table 1. Std. of noise of homomorphic filtering is 32.5, while the Std. of noise of proposed enhancement is 12.3. The experiment proves that the proposed enhancement is more effective in denoising compare with previous enhancement in high noise condition.

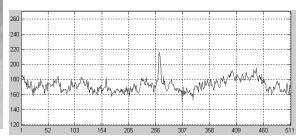
(a) Original image



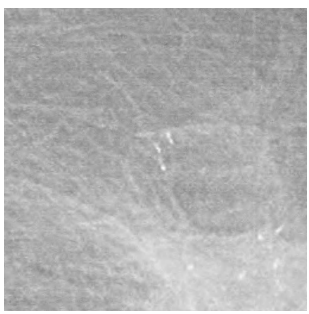
(b) Unsharp masking



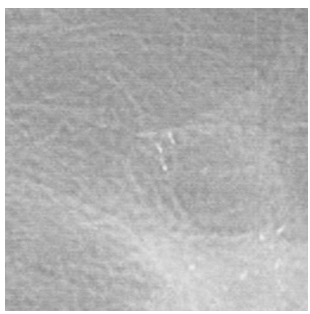
(e) Profile of (a)



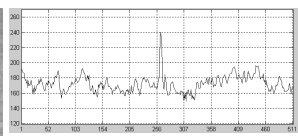
(f) Profile of (b)



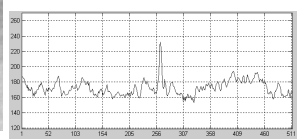
(c) Homomorphic filtering



(d) Proposed enhancement

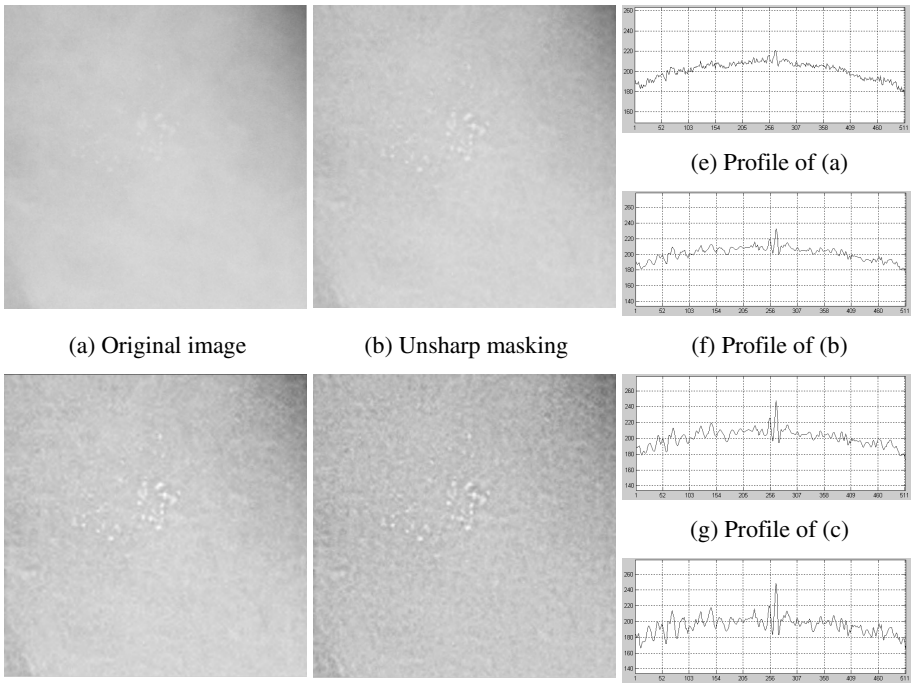


(g) Profile of (c)



(h) Profile of (d)

Fig. 6. Contrast enhancement for high noise mammography image and profile of one line



(a) Original image (b) Unsharp masking (e) Profile of (a) (f) Profile of (b)

(c) Homomorphic enhancement (d) Proposed enhancement (g) Profile of (c) (h) Profile of (d)

Fig. 7. Contrast enhancement for low noise mammography image and profile of one line

Table 1. Contrast Improvement Index and standard deviation of noise for high noise mammogram (Fig. 6)

	Original Image	Unsharp Masking	Homomorphic Filtering	Proposed Enhancement
C	0.0490	0.0871	0.1310	0.1103
CII	-	1.7761	2.6714	2.2500
Std. of noise	41.4	40.04	32.5	12.3

Table 2. Contrast Improvement Index and standard deviation of noise for low noise mammogram (Fig. 7)

	Original Image	Unsharp Masking	Homomorphic Filtering	Proposed Enhancement
C	0.0183	0.0382	0.0600	0.0710
CII	-	2.0906	3.2860	3.8858
Std. of noise	23.1	11.3	11.5	10.3

Fig. 7 provides another examples, in which original image has low noise. Fig. 7 and Table 2 show that noise is reduced in the all enhanced images. However, the proposed enhancement increases the contrast of microcalcification better than the others. In Table 2, the proposed enhancement obtains 3.8858 of CII while homomorphic filtering gets 3.2860. This example indicates that in low noise condition, proposed enhancement is better than the homomorphic filtering in contrast enhancement with similar denoising. This is due to the higher gains in the high frequency channels.

In conclusion, high noise mammograms have much fluctuation in breast region shown in profile of original image (Fig. 6 (e)). Unshap masking and homomorphic filtering enhance contrast of microcalcification with relatively high noise. On the other hand, proposed enhancement modifies wavelet gains and increases denoising threshold in high noise cases. Therefore, noise is reduced in breast area with high CII value. In low noise cases, proposed enhancement superior to other methods in denoising and improving contrast of microcalcifications.

5 Conclusion

In this paper, we propose a robust contrast enhancement method for microcalcification. The proposed method estimates noise characteristic in background region and eliminates noise in breast area incorporation with contrast enhancement of microcalcification. Experimental results show that, the proposed enhancement significantly reduces noise in high noise mammograms. In low noise mammograms, contrast of microcalcification is increased while noise is decreased.

Acknowledgment. Images were provided by University of South Florida DDSM (Digital Database for Screening Mammography). This paper is supported by the development of digital CAD system (02-PJ3- PG6-EV06-0002) of Ministry of Health and welfare in Republic of Korea.

References

1. Wang, T.C., Karayiannis, N.B.: Detection of Microcalcifications in Digital Mammograms Using Wavelets. IEEE Trans. Med. Imag., Vol. 17, No. 4, August (1998) 498–509.
2. Laine, A., Fan, J., Yang, W.: Wavelets for Contrast Enhancement of Digital Mammography. IEEE Engineering in Medicine and Biology Magazine, Vol.145. (1995) 536–550.
3. Yoon, J.H. Ro, Y.M.: Enhancement of the Contrast in Mammographic Images Using the Homomorphic Filter Method. IEICE Transactions on Information and Systems, Vol.85-D, No.1. (2002) 291–297.
4. Karssemeijer, N.: A stochastic model for automated detection of calcifications in digital mammograms. Proc. 12th Int. Conf. Information Processing Medical Imaging, Wye, U.K., (1991) 227–238.
5. Strickland, R.N., Hahn, H.I.: Wavelet transform for detecting microcalcifications in mammograms. IEEE Trans. Med. Imag., vol. 15. (1996) 218–229.

6. Yu, S., Guan, L.: A CAD System for the automatic Detection of Clustered Microcalcifications in Digitized Mammogram Film. IEEE Trans. Med. Imag., vol. 19,No 2. (1998) 115–126.
7. Sakellaropoulos, P., Costaridou, L., Panayiotakis, G.: A wavelet-based spatially adaptive method for mammographic contrast enhancement. Phys. Med. Biol. 48 (2003) 787–803.
8. Chang, S.G., Yu, B., Vetterli, M.: Adaptive Wavelet Thresholding for Image Denoising and Compression. IEEE Trans. Image Processing, Vol. 9, No. 9, (2000) 1532–1546.
9. Zheng, B., Qian, W., Clarke, L.P.: Digital mammography: Mixed feature neural net-work with spectral entropy decision for detection of microcalcifications. IEEE Trans. Med. Imag., vol. 15. (1996) 589–597.
10. Yoshida, H., Zhang, W., Cai, W., Doi, K., Nishikawa, R.M., Giger, M.L.: Optimizing wavelet transform based on supervised learning for detection of microcalcifications in digi-tal mammograms. Proc. IEEE ICIP, vol. 3, Washington, DC, (1995) 152–155.
11. Misiti, M., Misiti, Y., Oppenheim, G., Poggi, J.M.: Wavelet Toolbox User's Guide. Math Works Inc., Massachusetts, (1996).

Exact and Approximate Algorithms for Two–Criteria Topological Design Problem of WAN with Budget and Delay Constraints

Mariusz Gola[1] and Andrzej Kasprzak[2]

[1] Technical University of Opole, Department of Electrical Engineering and Automatic Control, Sosnkowskiego 31, 45-233 Opole, Poland
mariusz.gola@wp.pl
[2] Wroclaw University of Technology, Chair of Systems and Computer Networks, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
andrzej.kasprzak@pwr.wroc.pl

Abstract. This paper studies the problem of designing wide area networks (WAN). In the paper the two-criteria topology assignment problem with two constraints is considered. The goal is select flow routes, channel capacities and network topology in order to minimize the total average delay per packet and the leasing cost of channels subject to the budget constraint and delay constraint. The problem is NP-complete. Then, the branch and bound method is used to construct the exact algorithm. Also the approximate algorithm is presented. Some computational results are reported. Based on computational experiments, several properties of the considered problem are formulated.

1 Introduction

Topology design of the wide area network (WAN) is one of the most important functions a data engineer or manager can perform. Proper topology design of WAN saves money, time and resources. The data engineer must design of the WAN topology and anticipate change in user and technology environments. Therefore, an ongoing process of redesign and optimization is a prerequisite for an efficient and cost-effective WAN. Several different formulations of the topology design problem can be found in the literature; generally, they correspond to different choices of performance measures (criteria), of design variables and of constraints [1], [2], [3], [4], [5], [6]. Typically, the following performance measures are used: average delay, cost, throughput and network reliability. Also, the linear combinations of these performance measures are used in many applications. In many papers, the sum of the cost and average delay is used as criterion. To present such criterion function in the unified way, the following two distinct types of costs are considered: the leasing cost (cost of leasing of the channel capacities) and the delay cost (total average delay per packet times unit cost of delay). The introduced criterion is the sum of the leasing cost and the delay cost and it is called the combined cost. First, the combined cost criterion was proposed in the paper [3]. Then, the topology design problem with the combined cost criterion and with budget and delay constraints may by formulated as follows:

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 611–620, 2004.
© Springer-Verlag Berlin Heidelberg 2004

given: node location, possible channels location, channel capacity options and costs, maximal admissible average delay in WAN, budget of the WAN
minimize: combined cost
over: network topology, channel capacities, routing (i.e. multicommodity flow)
subject to: multicommodity flow constraints, channel capacities constraints, delay constraint, budget constraint.

We assume that channels capacities can be chosen from discrete sequence defined by ITU-T (International Telecommunications Union – Telecommunications Sector) recommendations. In this case the topology design problem is NP-complete [5].

The literature focusing on the topology design problem is very limited. At first, only the simple capacity and flow assignment (CFA) problems with combined cost criterion were considered in the literature [3], [5], [7], [8]. Next, the topology design problem was presented in the paper [9]. Of course, the topology design problem is more general than CFA problems considered in [3], [5], [7], [8] because there are in topology design problem more design variables (topology, channels capacities, flow) than in the CFA problem (channels capacities, flow). In turn, the problem considered in this paper is more general than the problem considered in the paper [9] because in this paper we take into account one constraint (budget constraint) more than in [9]. It causes that the problem considered here is more complicated and more hard to solve than the problem considered in the paper [9]. Moreover, conclusions presented here are more suitable for design process of WAN than conclusions presented in [9].

2 Problem Formulation

Consider a WAN with n nodes and b potential channels which may be used to build the network. For each potential channel i there is the set $\overline{ZC}^i = \{c_1^i, c_2^i, ..., c_{r(i)-1}^i\}$ of alternative capacities from which exactly one must be chosen if the i-th channel was chosen to build the WAN. Let d_k^i be the cost of the leasing capacity c_k^i [\$/month]. Let $c_{r(i)}^i = 0$ for $i = 1,...,b$. Then $ZC^i = \overline{ZC}^i \cup \{c_{r(i)}^i\}$ is the set of alternative capacities from among which exactly one must be used to channel i. If the capacity $c_{r(i)}^i$ is chosen then the i-th channel is not used to build the WAN. Let x_k^i be the decision variable which is equal to one if capacity c_k^i is assigned to channel i and x_k^i is equal to zero otherwise. Let $W^i = \{x_1^i, ..., x_{r(i)}^i\}$ be the set of all variables x_k^i which correspond to i-th channel. Since exactly one capacity from the set ZC^i must be chosen for channel i, then the following condition must be satisfied.

$$\sum_{k=1}^{r(i)} x_k^i = 1 \quad \text{for} \quad i = 1,...,b. \tag{1}$$

Let X_r' be the permutation of values of all variables x_k^i, $k = 1,...,r(i)$, $i = 1,...,b$ for which the condition (1) is satisfied, and let X_r be set of variables equal to one in X_r'. X_r is called a selection. Let $\Re$ be the family of all selections. The selection X_r defines the unique WAN because X_r determines simultaneously potential channels which are used to build the WAN (i.e. topology) and capacities for these channels; if $x_k^i \in X_r, k < r(i)$, then the channel i is used to build the WAN and capacity of this channel is equal to c_k^i; if $x_{r(i)}^i \in X_r$ then the channel i is not used to build WAN.

Let $T(X_r)$ be the minimal average delay per packet in the WAN with topology and channel capacities given by the selection X_r. The average delay per packet expression is given by Kleinrock's formula [4]. $T(X_r)$ can be obtained by solving a multicommodity flow problem in the network with topology and channel capacities given by X_r. Let $d(X_r)$ be the sum of leasing costs of channel capacities given by X_r and let a be the unit cost of delay. Then, the combined cost is following:

$$Q(X_r) = a \cdot T(X_r) + d(X_r).$$

Let $T_{\max}$ be the maximal admissible average delay per packet in WAN and let B be the budget (maximal feasible leasing capacity cost) of the WAN.

Then, the topology design problem with the combined cost criterion and with budget and delay constraints can be formulated as follows:

$$\min_{X_r} Q(X_r) \tag{2}$$

subject to

$$X_r \in \Re \tag{3}$$

$$d(X_r) = \sum_{x_k^i \in X_r} x_k^i d_k^i \le B \tag{4}$$

$$T(X_r) \le T_{\max} \tag{5}$$

3 The Branch and Bound Algorithm

The detailed description of the calculation scheme of the branch and bound method may be found in the paper [10]. Starting with initial selection X_1 we generate a sequence of selections. The variable x is called normal if $x \in X_1$. The variable which does not belong to X_1 is called reverse. A replacement of any variable by other variable from W^i is called complementing. The generation of new selection involves the choice of a certain normal variable from X_r for complementing; it is called the

branching rules. The choice of the normal and reverse variables is based on local optimization criterion. For each selection X_r we calculate lower bound to check the possibility of the generation of the selection X_s with less combined cost than already found. If such X_s does not exist, we abandon X_r and all its possible successors and next backtrack to the predecessor from which X_r was generated. If the new selection X_s is generated from X_r by complementing the normal variable by the reverse variable, then we constantly fix the reverse variable. It means that this reverse variable cannot be complemented by any other in every possible successor of X_s. If we backtrack from X_s to X_r by the reverse variable of certain normal variable from X_r we momentarily fix this normal variable. So, for each X_r we constantly fix a set F_r and momentarily fix a set F_r^t. The reverse variables in F_r are constantly fixed. Each momentarily fixed variable in F_r^t is the reverse variable abandoned during backtracking process. The variables which do not belong to F_r or F_r^t are called free in X_r. If we want backtrack from X_1 then the algorithm terminates.

3.1 Branching Rules

The purpose of the branching rules is to find the normal variable from the selection X_r for complementing and generating a successor of X_r. If the conditions (4) and (5) are satisfied for the selection X_r, then the choice of the normal variable for complementing is performed using the *choice operation*, else this choice is performed using the *regulation operation*.

Choice operation. The basic task of this operation is to choose the normal and reverse variables for complementing and generating a successor with the least possible combined cost.

Let γ be the total average packet rate from external sources, and let ε be the minimal feasible difference between capacity and flow in each channel.

Theorem 1. Let $X_r \in \Re$. If the selection X_s is obtained from X_r by complementing the variable $x_k^i \in X_r$ by the variable $x_j^i \in X_s$ then $Q(X_s) \le Q(X_r) - \Delta_{kj}^{ir}$, where

$$\Delta_{kj}^{ir} = \begin{cases} \dfrac{a}{\gamma}\left(\dfrac{f_r^i}{c_k^i - f_r^i} - \dfrac{f_r^i}{c_j^i - f_r^i}\right) + \left(d_k^i - d_j^i\right) & \text{if} \quad c_j^i \ge f_r^i \\ \dfrac{a}{\gamma}\left(\dfrac{f_r^i}{c_k^i - f_r^i} - \dfrac{1}{\varepsilon}\left(\max\limits_{x_l^i \in W^i - F_r^t} c_l^i\right)\right) + \left(d_k^i - d_j^i\right) & \text{otherwise} \end{cases} \tag{6}$$

and f_r^i is the flow in the *i*-th channel obtained by solving the multicommodity flow problem for network topology and channel capacities given by the selection X_r.

Let $E_r = X_r - F_r$, and let M_r be the set of all reverse variables of normal variables which belongs to the set E_r. We want to choose a normal and reverse variable for complementing which generates a successor with the possibly least combined cost. It follows from theorem 1 that we should choose such normal variable $x_k^i \in E_r$ and reverse variable $x_j^i \in M_r$ for which the value of Δ_{kj}^{ir} is maximal.

Regulation operation. It is possible that we obtain the selection X_s for which at least one of conditions (4) and/or (5) are not satisfied after complementing the normal variable from the selection X_r by the reverse variable from the set M_r. Then, one of the three following cases is true. In each case we use the different local optimization criterion.

Case A (condition (4) is not satisfied and condition (5) is satisfied).
Corollary 1. If the selection X_s is obtained from X_r by complementing normal variables from X_r and if $d(X_s) < d(X_r)$, then X_s contains at least one reverse variable x_j^i of the normal variable $x_k^i \in X_r$ such that $d_j^i < d_k^i$.

Let $P_r \subset M_r$ be the set of reverse variables which satisfy the following condition: if $x_j^i \in P_r$ and $x_k^i \in E_r$ then $d_k^i > d_j^i$. It follows from Corollary 1 that we may complement only variables belonging to the set P_r. The choice criterion on variables from P_r is the value $\theta^d = \Delta_{kj}^{ir} \cdot \left(\tilde{T}(X_s) - T(X_r)\right)^{\kappa}$, where $\tilde{T}(X_s)$ is upper bound of total average delay per packet in the WAN in which network topology and channel capacities are given by the selection X_s; $\kappa = -1$ if $\Delta_{kj}^{ir} \geq 0$ and $\kappa = 1$ otherwise.

Case B (condition (4) is satisfied and condition (5) is not satisfied).
Corollary 2. If the selection X_s is obtained from X_r by complementing normal variables from X_r and if $T(X_s) < T(X_r)$, then X_s contains at least one reverse variable x_j^i of the normal variable $x_k^i \in X_r$ such that $c_j^i > c_k^i$.

Let $P_r' \subset M_r$ be the set of reverse variables which satisfy the following condition: if $x_j^i \in P_r'$ and $x_k^i \in E_r$ then $c_j^i > c_k^i$. It follows from Corollary 2 that we may complement only variables belonging to the set P_r'. The choice criterion on variables from P_r' is the value $\theta^T = \Delta_{kj}^{ir} \cdot \left(d(w_j^i) - d(w_k^i)\right)^{\kappa}$.

Case C. (both conditions (4) and (5) are not satisfied). In this case we chose normal variable from the set E_r and reverse variable from the set M_r. We should choose such normal and reverse variables for which the value of expression (6) is maximal.

3.2 Lower Bound

The lower bound LB_r of minimal value of the combined cost for every possible successor X_s generated from selection X_r may be obtained by relaxing or omitting some constraints in the problem (2-5). It is easy to observe, if we omit the constraint (4) in our problem (2-5) then we obtain the problem considered in [9]. Then the lower bound introduced and proved in the paper [9] may be applied for the problem (2-5).

4 Approximate Algorithm

The presented exact algorithm involves the initial selection $X_1 \in \Re$ for which the constraints (4) and (5) are satisfied [10]. Moreover, the initial selection should be near-optimal solution of the problem (2-5). To find the initial selection the following approximate algorithm is proposed. Of course, this approximate algorithm may be also used to design of the WAN when the optimal solution is not necessary.

Step 1. Assign maximum available capacities to channels, i.e. perform $r = 1$ and $X_1^{(r)} = \{x_1^1, x_1^2, ..., x_1^b\}$. Next compute $T(X_1^{(1)})$. If $T(X_1^{(1)}) > T_{\max}$ then algorithm terminates – the problem (2-5) has no solution. Otherwise perform $Q^* = \infty$ and $S^{(1)} = \bigcup_{i=1}^{b} W^i - X_1^{(1)}$. Next go to step 2.

Step 2. If $S^{(r)} = \varnothing$ then go to step 6. Otherwise go to step 3.

Step 3. Choose the variable $x_k^i \in X_1^{(r)}$ and the variable $x_j^i \in S^{(r)}$ for which the value $((d_k^i - d_j^i)/(\tilde{T}(X_1^{(r+1)}) - T(X_1^{(r)})))$ is maximal. Next perform $S^{(r)} = S^{(r)} - \{x_j^i\}$. Generate the new selection $X_1^{(r+1)} = (X_1^{(r)} - \{x_k^i\}) \cup \{x_j^i\}$. Compute $T(X_1^{(r+1)})$. If $T(X_1^{(r+1)}) > T_{\max}$ then go to step 2. Otherwise go to step 4.

Step 4. If $d(X_1^{(r+1)}) > B$ then go to step 5. Otherwise compute $Q(X_1^{(r+1)})$. If $Q(X_1^{(r+1)}) > Q^*$ then go to step 2. Otherwise perform $Q^* = Q(X_1^{(r+1)})$ and go to step 5.

Step 5. Perform $S^{(r+1)} = S^{(r)} - \bigcup_{l=j+1}^{r(i)-1} \{x_l^i\}$, $r = r + 1$ and go to step 2.

Step 6. If $T(X_1^{(r)}) \le T_{\max}$ and $d(X_1^{(r)}) \le B$ then perform $X_1 = X_1^{(r)}$ and algorithm terminates. The feasible solution was found. Otherwise the problem (2-5) has no solution.

5 Computational Results

The presented exact and approximate algorithms were implemented in C++ code. Extensive numerical experiments have been performed with these algorithms for many different network topologies. The experiments were conducted with two main purposes in mind: first, to test the computational efficiency of the algorithms and second, to examine the impact of various parameters on solutions to find possible properties of the considered problem.

In order to evaluate the effectiveness of the presented exact algorithm from the computational point of view, it was applied to many networks and the obtained results (i.e. number of iterations of the algorithm) were compared.

Let $D_{\max}$ be the maximal building cost of the network, and let $D_{\min}$ be the minimal building cost of the network; the problem (2-5) has no solution for $B < D_{\min}$. To compare the results obtained for different networks topologies we introduce the normalized budget $NB = ((B - D_{\min})/(D_{\max} - D_{\min})) \cdot 100\%$.

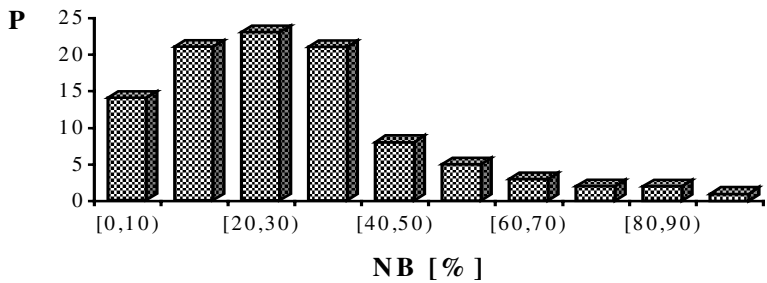


Fig. 1. The dependence of *P* on normalized budget *NB*

Moreover, let $P^i(NB)$ be the number of iterations of the branch and bound algorithm to obtain the optimal value of *Q* for normalized budget equal to *NB* for *i*-th considered network topology. Let

$$P(u,v) = \frac{1}{Z}\sum_{i=1}^{Z}\left(\sum_{NB\in[u,v]} P^i(NB) \Big/ \sum_{NB\in[1,100]} P^i(NB) \right) \cdot 100\%$$

be the arithmetic mean of the relative number of iterations for $NB \in [u,v]$ calculated for all considered network topologies, where *Z* is the number of considered wide area network topologies. Fig. 1 shows the dependency of *P* on divisions [0%,10%),

[10%,20%),...,[90%,100%] of normalized budget *NB*. It follows from Fig. 1 that algorithm is especially effective from computational point of view for $NB \in [40,100]$.

The quality of the approximate algorithm was examined. Let the distance between approximate and optimal solutions be denoted by $k = \left(\left|Q(X_1) - Q\right| / Q\right) \cdot 100\%$. The value *k* shows how approximate solutions are worse than optimal solutions. Let

$$K[u,v] = \frac{\text{number of solutions for which } k \in [u,v]}{\text{number of all solutions}} \cdot 100\%$$

denotes number of solutions obtained from approximate algorithm (in percentage) which are greater than optimal solutions more than *u%* and less than *v%*. In the Fig. 2 the distance between approximate and optimal solutions are presented.

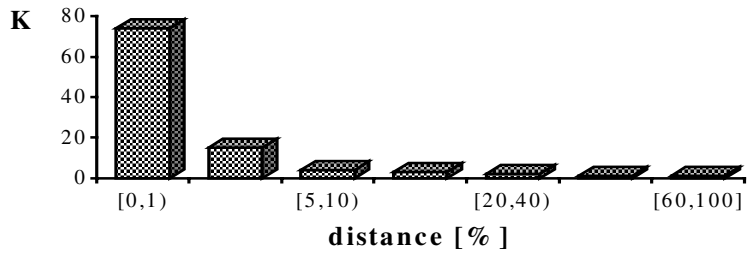


Fig. 2. The distance between the approximate and optimal solutions

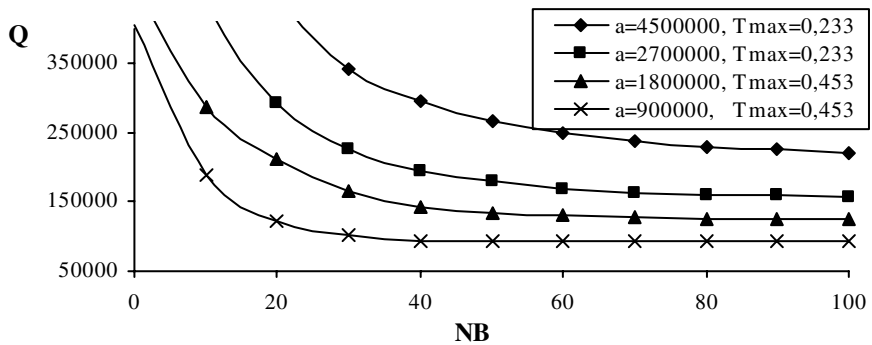


Fig. 3. The dependence of the criterion function *Q* on normalized budget *NB*

The dependence of the optimal combined cost *Q* on budget *B* has been examined. In the Fig. 3 the typically dependence of combined cost *Q* on normalized budget *NB* is presented for different values of the parameter *a*. It follows from Fig. 3 that there exists such budget B^*, that the problem (2-5) has the same solution for each *B* greater or equal to B^*. It means that the optimal solution of the problem (2-5) is on the budget constrains (4) for $B \le B^*$, and it is inside the set of feasible solution for $B > B^*$. Moreover, this observation is very important from practical point of view. It

shows that the influence of the investment cost (budget) on the optimal solution of the considered topology design problem is limited.

Conclusion 1. In the problem (2-5), for fixed $T_{\max}$, there exists such value B^* of the budget B that for each $B \geq B^*$ we obtain the same optimal solution. It means that for $B \geq B^*$ the constraint (4) may be substituted by constraint $d(X_r) \leq B^*$.

Let T^{opt} be the average delay per packet in the WAN and let d^{opt} be the cost of build of the WAN obtained by solving the problem (2-5). In other words, if the value of criterion function Q is minimum and the constrains (3-5) are satisfied then $Q = a \cdot T^{opt} + d^{opt}$.

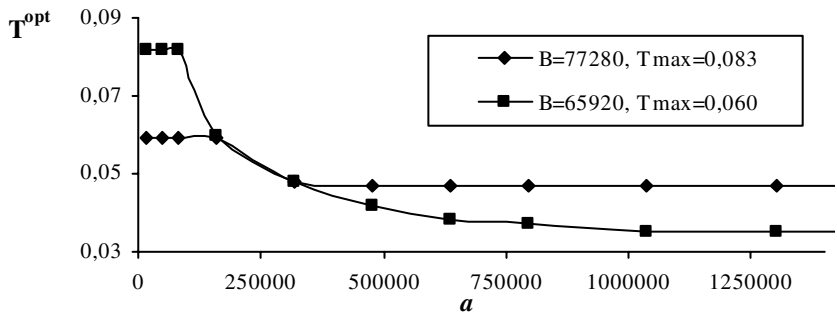


Fig. 4. Dependence of T^{opt} on coefficient a

The dependences of the average delay per packet T^{opt} and building cost d^{opt} on coefficient a have been examined. In the Fig. 4 the typically dependence of T^{opt} on a is presented for different values of maximal admissible average delay per packet $T_{\max}$ and different values of budget B. In the Fig. 5, the typically dependence of d^{opt} on a is presented for different values of maximal average delay per packet $T_{\max}$ and different values of budget B. The observations following from these computer experiments may be formulated in the form of the below conclusions.

Conclusion 2. In the problem (2-5), there exists such value $\hat{a}$ of coefficient a that for every $a > \hat{a}$ the following inequality is true: $T(X_{opt}) < T_{\max}$, where X_{opt} denotes the optimal solution of the problem (2-5).

Conclusion 3. In the problem (2-5), there exists such value a'' of coefficient a that for every $a < a''$ the following inequality is true: $d(X_{opt}) < B$.

Conclusion 4. There exists such range $(\hat{a}, a'')$ that for every $a \in (\hat{a}, a'')$ the problem (2-5) may by solved without the budget constraint (4) and without the delay constraint (5). It means that the optimal solution of the problem (2-3) is inside the set of feasible solutions for the problem (2-5) and it is also optimal solution of the problem (2-5).

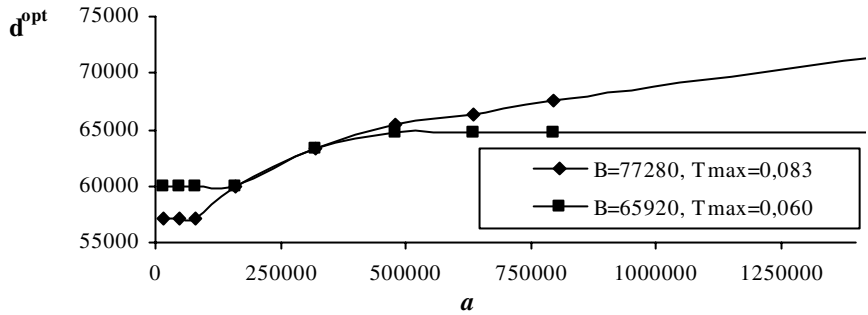


Fig. 5. Dependence of d^{opt} on coefficient a

6 Conclusion

The exact and approximate algorithms for solving topology design problem in WAN are presented. The considered problem is more general than the similar problems presented in the literature. It follows from computational experiments (Fig. 2) that about 75% approximate solutions differ from optimal solutions at most 1%. Moreover, presented conclusions allow to simplify the design process of the WAN.

References

1. Alevras D., Groetschel M., Weesaly R.: Cost Efficient Network Synthesis from Leased Lines. Annals of Operation Research 76 (1998) 1–20
2. Drakopoulos E.: Enterprise network planning and design: methodology and application. Computer Communications 22 (1999) 340–352
3. Gavish B., Neuman I.: A System for Routing Capacity Assignment in Computer Communication Networks. IEEE Transactions on Communication. 37 (1989) 360–366
4. Gerla M., Kleinrock L.: On the Topological Design of Distributed Computer Network. IEEE Transactions on Communication 25 (1977) 48–60
5. Kasprzak A.: Topological Design of the Wide Area Networks. Wroclaw University of Technology Press, Wroclaw (2001)
6. Walkowiak K.: A New Approach to Survivability of Connection Oriented Networks. Lecture Notes in Computer Science 2657 (2003) 501–510
7. Gola M., Kasprzak A.: The Capacity and Flow Assignment in Wide Area Computer Networks: An Algorithm and Computational Results, Proc. 15th IMACS World Congress on Scientific Computation, Modeling and Applied Mathematics, Berlin (1997) 585–590
8. Gola M., Kasprzak A.: An Exact Algorithm for CFA problem in WAN with Combined Cost Criterion and with Budget and Delay Constraints. Proc. Sixteenth European Meeting on Cybernetics and System Research, Vienna (2002) 867–872
9. Gola M., Kasprzak A.: The Two-Criteria Topological Design Problem in WAN with Delay Constraint: An Algorithm and Computational Results. Lecture Notes in Computer Science 2667 (2003) 180–189
10. Wolsey L. A.: Integer Programming. Wiley-Interscience (1998)

Data Management with Load Balancing in Distributed Computing

Jong Sik Lee

School of Computer Science and Engineering
Inha University
Incheon 402-751, South Korea
jslee@inha.ac.kr

Abstract. This paper reviews existing data management schemes and presents a design and development of a data management scheme with load balancing in a distributed computing. This scheme defines a variety of degree of load balancing, maps each degree to each data management configuration, and reduces data traffic among distributed components. The scheme allows to share geographically dispersed data assets collaboratively and execute a complex large-scale distributed computing system including cooperating and distributed components with reasonable computation and communication resources. In addition, this paper introduces a HLA (High Level Architecture) bridge middleware environment for data communication among multiple federations. We analyze system performance and scalability with a variety of degree of load balancing configurations. The empirical result on a heterogeneous OS distributed system apparently presents advantages of the data management scheme with load balancing in terms of system performance and scalability.

1 Introduction

Distributed computing is being noticed for a growing variety of systems including process control and manufacturing, military command and control, transportation management, and so on. Such distributed computing systems are complex and large in their size. A large-scale distributed computing system requires an achievement of real-time linkage among multiple and geographically distant systems, and thus has to execute complex large-scale execution and to share geographically dispersed data assets and computing resources collaboratively. However, large-scale distributed computing systems are characterized by numerous interactive data exchanges among components distributed between computers networked together. The methodology to support the reduction of the interactive messages among distributed computing components is called "data management." In this paper, we propose a data management scheme with load balancing to promote the effective reduction of data communication in a distributed computing environment.

The data management scheme with load balancing classifies the degree of load balancing: low and high. The classification shows the degree of performance improvement with varying degree of load balancing. The data management with load balancing is extended from previously developed data distribution management

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 621–629, 2004.
© Springer-Verlag Berlin Heidelberg 2004

schemes [1, 2] to execute a complex and large-scale distributed system with reasonable computation and communication resources. In addition, a bridge-based inter-federation communication system with the HLA [3, 4, 5] middleware is provided to support the communication among distributed components.

This paper is organized as follows: Section 2 reviews existing data management schemes. Section 3 discusses the data management with load balancing on an application: satellite cluster management. Section 4 analyzes performance effectiveness and system scalability. Section 5 illustrates a testbed for experiment and evaluates system performance with low and high load balancing approaches. The conclusion is Section 6.

2 Data Management Schemes

This section briefly overviews of the major data management schemes, which are currently used in most entity-based virtual simulations. These schemes include dead-reckoning, interest management, and Data Distribution Management (DDM) of HLA.

2.1 Dead-Reckoning

As a scheme to reduce the number of state update messages, the dead-reckoning scheme [6, 7] is widely employed in distributed simulations. The state update messages are exchanged among each simulated entity to maintain the accurate state of the other remote simulated entities. Each federate maintains accurate information (position, velocity, acceleration) of its own simulated entity's movement with a high fidelity model. Also, each federate includes the dead-reckoning (inaccurate) models of all simulated entities including that of its own entity.

2.2 Interest Management

The interest management technique [8] was proposed as a method to avoid broadcast communication among agents. Generally, the interest management technique is a message filtering mechanism to enable execution with the reasonable communication and computation resources in real-time large-scale simulations. Interest management is based on interest expression between pairs of sender and receiver agents. The receiver agent expresses the interest to an attribute of the sender agent and the sender agent sends the value of the attribute interested to the receiver agent.

2.3 Data Distribution Management (DDM) of HLA

HLA provides the DDM service as an example of the interest management. In the DDM, the interest expression works with regions in a multi-dimensional parameter space. The multi-dimensional coordinate system is call the "routing space" and the routing space is subdivided into a predefined array of fixed sized cells. Each cell is assigned to a multicast group [4]. The DDM [4, 9, 10] service of HLA constitutes an

interest-based message traffic reduction scheme. This service tries to filter out irrelevant data among federates. Each federate expresses the interest for the data to be sent and received by defining publication region and subscription region in the routing space.

3 Data Management with Load Balancing

Data management is a hot issue to improve the performance a complex and large-scale distributed computing system since the data management reduces data transmission among distributed components. This paper proposes a data management scheme with load balancing which reduces the required transmission data by assigning different communication and computation load to each transmission-related component. Especially, the proposed scheme focuses on *communication load balancing* which indicates communication load distribution to each distributed component from a centralized communication-related component. This scheme reduces system execution cost by reducing the communication cost among components with separated communication loads. In this paper, we introduce a satellite cluster management system as an application and apply to the data management scheme with load balancing to the system.

3.1 Application: Satellite Cluster Management System

Separated spacecrafts in a satellite cluster occupies their distributed space assets. The management of the distributed characters is essential to progress the satellite cluster mission [11, 12] in the cluster functionalities such as resource management, navigation, guidance, fault protection, and so on. While a centralized management approach is defined, the cluster manager provides the cluster functionalities. The operation of the cluster manager consists of four categories: spacecraft command and control, cluster data management, flying formation and fault management.

We introduce the ground system operation as a case study to discuss non-load balancing and load balancing approaches and evaluate system performance. A ground system commands and controls a cluster of spacecrafts. Basically, a ground system requires operations and manpower to monitor the cluster, makes a decision, and sends the proper command strings. For a small cluster, a centralized approach is cost effective and accepted to command and control spacecrafts individually.

To improve the total system performance by reducing the required transmission data, the load balancing approach of ground operations is proposed in this paper. The load balancing approach indicates that it separates ground functions and distributes a set of functions to spacecrafts. Performing a set of functions requires communication loads of spacecraft. Here, we classify the degree of load balancing: low and high. Fig. 1 illustrates the low load balancing. A ground station separates four regions to be observed, makes four different command strings, and sends them to a cluster manager. The cluster manager parses the command strings and forwards them to each proper spacecraft. The parsing and forwarding classifies light loads of the cluster manager. The cluster manager includes the light loads, however the heavy communication data are required between the cluster manager and the ground station.

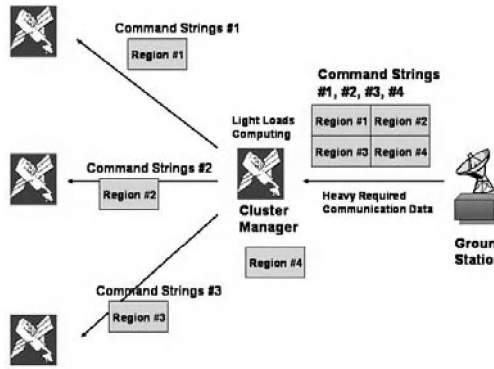


Fig. 1. Low Load Balancing Approach

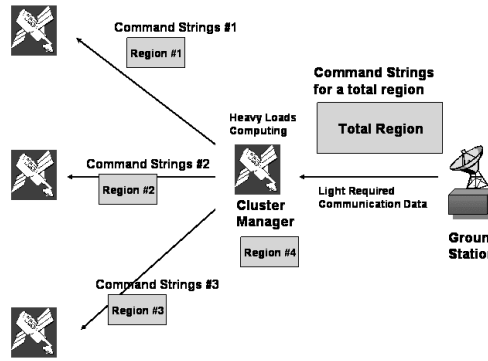


Fig. 2. High Load Balancing Approach

Fig. 2 illustrates the high load balancing with the cluster manager over parsing and forwarding. The ground station does not separate four regions to be observed and sends a total region to the cluster manager. The cluster manager should include the load for division of region to be observed. The division load should understand the technology including region division, image capturing, image visualization, image data transmission, and so on. The cluster manager includes the heavy loads, however the light communication data are required between the cluster manager and the ground station.

4 Performance Analysis

To analyze the system performance of the data management with load balancing, we take the amount of required satellite transmission data which are between ground station and spacecrafts. Notice that transmission data among spacecrafts inside cluster is ignored.

Table 1. Analysis of Transmission Data Reduction (Note: N: Number of spacecrafts in a Cluster; M: Number of Clusters; H: Number of overhead bits in satellite communication (160 bits assumed), R: Number of regions at one spacecraft on one transmission (40 assumed))

Approach		Number of transmission required messages	Number of bits transmitted
Non-Load Balancing		R*N*M	(H + 4*64) *R*N*M
Degree of Load Balancing	Low	M	(H + 4*64*R*N) *M
	High	M	(H + 4*64) *M

As Table 1 shows, the load balancing approach significantly reduces the number of messages passed and the number of bits passed. Basically, there occurs overhead bits (H) needed for satellite communication when a ground station sends a command. The centralized approach causes an amount of overhead messages and bits since it makes a ground station to send spacecrafts messages individually. High load balancing significantly reduces the transmission data bits since it transmits the one big region location information irrelevant to the number of spacecrafts (N) in a cluster. Specially, as the number of spacecrafts (N) goes infinity, the requirement of transmission data bits in the low load balancing increases linearly. The increasing slope is (4*64)*M. However the high load balancing still requires the same lower transmission data bits. The analysis in Table 1 reveals that, especially the large numbers of spacecrafts working in a cluster, we expect the greatest transmission data reduction in the decentralized approach with the high load cluster manager. In the decentralized approach, the computation overhead for execute the load. However we can ignore the computation overhead since communication resource is a critical factor to execute a satellite system within reasonable time and current computation technology is already high and grows fast.

5 Experiment and Performance Evaluation

5.1 Testbed Environment

To evaluate the performance of the data management with load balancing, we define a scenario of cluster operation. A cluster of 4 spacecrafts flies on pre-scheduled orbits. One of the spacecrafts acts as the cluster manager that communicates with the ground station. The cluster manager gathers states of each spacecraft and sends telemetry information back to the ground station. At any given time, the ground station can send an observation request to the cluster manager, which in turns, will coordinate with other spacecrafts in the cluster to perform the requested observation in synchronization. The cluster manager then aggregates data collected from the other spacecrafts and send it back to the ground station. To execute the scenario, we develop two testbeds for inside-federation and inter-federation communications. As Fig. 3 illustrates, the inside-federation communication works on a cluster system federation. The federation includes four spacecrafts federates, including cluster manager, and one ground station federation. The RTI message passing for cluster data

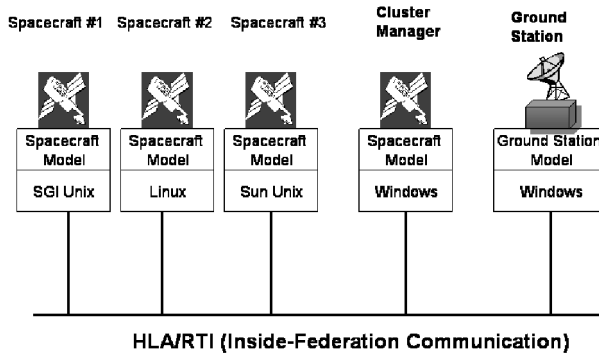


Fig. 3. Inside-federation communication system

management depends on the inside-federation communication. In the platform setting, we develop a heterogeneous distributed system which includes various operating systems including SGI Unix, Linux, Sun Unix, and Windows. The total of five federates are allocated to five machines, respectively, and they are connected via a 10 Base T Ethernet network.

For inter-federation communication with a bridge federate, we develop two federations: cluster and ground. The cluster federation includes four spacecraft federates, including cluster manager, and the ground federation includes two federates: cluster manager and ground station. Both federations have the cluster manager federate which is called bridge federate. The HLA Bridge implementation supports the bridge federate functionality for the inter-federation RTI message passing, thus it makes the inter-federation communication executable.

5.2 Performance Evaluation

In order to evaluate system execution performance of the data management with load balancing, we compare transmission data bits in three approaches: non-load balancing, low load balancing, and high load balancing. The compassion is achieved with variation of number of satellites. The non-load balancing approach is executed on only one federation which provides inside-federation communication as shown Fig. 3. The load balancing approach is executed on two federations: cluster and ground. As Fig. 4 illustrates, the load balancing approach apparently reduces the transmission data bits. Especially, the use of high load balancing approach greatly reduces the transmission data bits. The high load balancing approach is able to allow an execution which requires a small amount of transmission data bits regardless to the number of satellites.

We use system execution time as the other evaluation measure of system execution performance. The system execution time considers communication and computation performance. The non-load balancing approach requires a large amount of communication data, however it does not need the local computation for balanced load. The system execution time for the centralized approach is mostly caused from the amount of communication data. The load balancing approach reduces the amount of communication data and uses operations for load balancing. The system execution

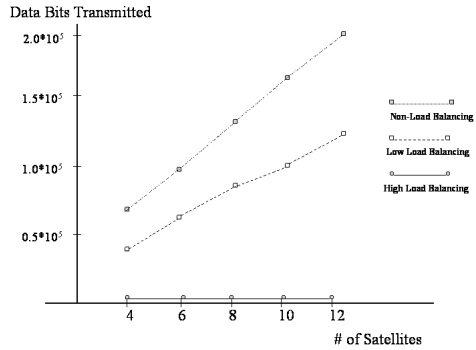


Fig. 4. Transmission data bits (Non-Load Balancing vs. Low Load Balancing vs. High Load Balancing)

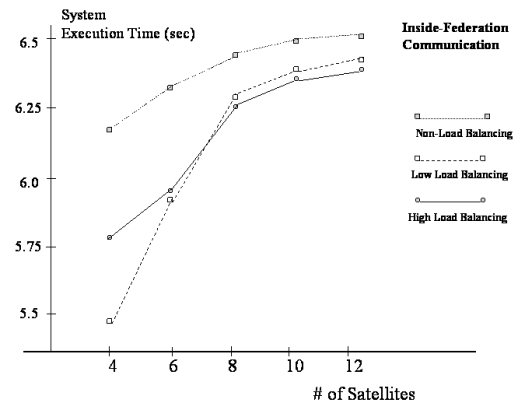


Fig. 5. System execution time on inside-federation communication system (Non-Load Balancing vs. Low Load Balancing vs. High Load Balancing)

time for the load balancing approach is caused by both of data communication time and load operation time. Especially, the high load balancing approach requires more load operation time that that for low load balancing. Fig. 5 compares system execution time in three approaches: non-load balancing, low load balancing, and high load balancing. The system execution time of Fig. 5 is provided from the execution on only one federation with inside-federation communication. The load balancing approach apparently reduces system execution time. The reduction indicates that the time reduction from transmission data reduction is greater than the time expense from load operation. In comparison between high and low load balancing, there exists a tradeoff between transmission data reduction and degree of load balancing. In inside-federation communication system of Fig. 5, the low load balancing shows the lower execution time in the lower task load. The smaller number of satellites presents the lower task load. As the task load increases, the high load balancing shows the lower execution time. We observe system execution time with inter-federation communication on the two federations. The non-load balancing approach is not existed since the approach cannot be operated with inter-federation communication.

The high load balancing shows the lower execution time in the lower task load. As the task load increases, the low load balancing makes its execution time increase, thus its execution time becomes closed to that of high load balancing.

6 Conclusion

This paper overviewed a variety of data management schemes and presented the design and development of the data management with load balancing in a distributed computing system. In addition, for practical construction and execution of the distributed computing system, we reviewed distributed system construction concepts, including functionality balancing, system robustness and maintainability. As noticed in this paper, the bridge-based inter-federation communication in HLA-compliant distributed computing improves the modeling flexibility by allowing multiple connections among distributed components which have a variety of topology. Thus, the modeling flexibility allows to analyze a complex large-scale distributed computing system and to get empirical results. The proposed scheme focuses on the different load balancing to each distributed component and assigns various degrees of communication and computation loads in each component. This load balancing approach, which is applied in data management among distributed components, allows a various complex execution for a variety of distributed computing systems and improves system performance through data communication reduction and local computation load balancing. We analyzed the system performance and scalability of the data management with load balancing. The empirical results showed favorable reduction of communication data and overall execution time and proved the usefulness in distributed computing system.

References

1. Jong S. Lee and Bernard. P. Zeigler: Space-based Communication Data Management in Scalable Distributed simulation. Journal of Parallel and Distributed Computing 62. (2002) 336–365
2. Jong Sik Lee, Bernard. P. Zeigler: Design and Development of Data distribution Management Environment. Journal of Society Computer Simulation, Simulation, Vol. 77. (2002) 39–52
3. Defense, D.o.: Draft Standard For Modeling and Simulation (M&S) High Level Architecture(HLA) - Federate Interface Specification, Draft 1. (1998)
4. High Level Architecture Run-Time Infrastructure Programmer's Guide 1.3 Version 3, DMSO (1998)
5. Nico Kuijpers, et al.: Applying Data Distribution Management and Ownership Management Services of the HLA Interface Specification. in SIW. Orlando FL (1999)
6. Lin, C.: Study on the network load in distributed interactive simulation. Proceeding of the AIAA on Flight Simulation Technologies (1994)
7. Lin, C.: The performance assessment of the dead-reckoning algorithm in DIS. Proceedings of the 10th DIS Workshop on Standards for the Interoperability of Distributed Simulation (1994)

8. Katherine L. Morse: Interest management in large scale distributed simulations. Tech. Rep. 96-127, Department of Information and Computer Science. University of California Irvine (1996)
9. Boukerche and A. Roy: A Dynamic Grid-Based Multicast Algorithm for Data Distribution Management. 4th IEEE Distributed Simulation and Real Time Application (2000)
10. Gary Tan, et. al.: A Hybrid Approach to Data Distribution Management. 4th IEEE Distributed Simulation and Real Time Application (2000)
11. P. Zetocha: Intelligent Agent Architecture for Onboard Executive Satellite Control. Intelligent Automation and Control, vol. 9. TSI Press Series on Intelligent Automation and Soft Computing, Albuquerque N.M. (2000) 27–32
12. D.M. Surka, M.C. Brito, and C.G. Harvey: Development of the Real-Time Object-Agent Flight Software Architecture for Distributed Satellite Systems. IEEE Aerospace Conf., IEEE Press, Piscataway N.J. (2001)

High Performance Modeling with Quantized System

Jong Sik Lee

School of Computer Science and Engineering
Inha University
Incheon 402-751, South Korea
jslee@inha.ac.kr

Abstract. As analyses of system behavior and complexity through computer modeling and simulation have been growing, high performance modeling is noticed to handle behavior and complexity of modern large-scale systems. The high performance modeling focuses on high resolution representation of system and performance improvement. This paper presents a quantized system modeling as an effective high performance modeling. The quantized system modeling represents state resolution using state quantization and improves system performance through message filtering among system components. This paper realizes a practical quantization component which is based on both of Discrete Time System (DTS) and Discrete Event System (DES) and shows usefulness of quantized system in a variety of industrial applications. This paper models a real-world application, e.g. a space traveling system with the quantized system and evaluates system performance, accuracy, and scalability of the quantized system on DES-based modeling and simulation.

1 Introduction

As system behavior and complexity analyzed by computer has been increasing, high performance modeling [1, 2, 3, 4, 5] is demanded to deal with behavior and complexity of large-scale systems. For high performance modeling, high-resolution and large-scale representations of system is needed to handle behavior of large-scale modern system. This paper presents a quantization-based system modeling of a complex and large-scale system to support high-resolution and large-scale representations. The quantized system modeling [6, 7, 8, 9, 10] is based on a quantization of state of system and provides a high performance simulation.

This paper reviews a discrete event-based system modeling specification, which is called DEVS (Discrete Event System Specification) [11, 12] and specifies an existing DTS in sense of strong representation of DEVS. For a realization of the quantized system modeling and DEVS, we present a quantized DEVS integrator [11] which provides behaviors and characteristics of discrete event-based system. To validate a fidelity of the quantized system modeling and DEVS, a kinetics of a spaceship is taken an application. The kinetics maintains an accounting of where ships are and predicting their future destinations. We describe a workable representation of the DTSS (Discrete Time System Specification) [11] formalism in DEVS. For performance evaluation, we model the kinetics on both of DTSS and DEVS formalisms and environments and compare system performance each other. Section 2

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 630–637, 2004.
© Springer-Verlag Berlin Heidelberg 2004

describes quantization approach and show how the quantization approach can be applied to improve system performance. Section 3 presents a quantized system and realizes a quantized DEVS integrator. Section 4 presents a modeling of a kinetics of spaceship. Section 5 discusses experiment and performance evaluation. Section 6 is conclusion.

2 Quantization and Performance

Quantization is based on the quantization theory [6, 7] with modeling formalism and system homomorphisms. A continuous trajectory with a finite number of values in a finite time interval is approximated. In order to obtain a discrete time system approximation, discretization of the time base is needed with a finite time interval. The finite number of values is then calculated from the partition of the trajectory into a finite number of segments (each of which has a finite computation). The partition of the trajectory with the finite number of values provides a way to quantize the value space, which is partitioned in every D interval (quantum), and the time space is partitioned in every T interval (time interval).

In discrete event systems, we sample the time values at every quantum interval (D), use discrete values with continuous time, and send the quantum levels out after the sampled time interval. This is called the quantization based on D. In a real application, the state trajectory is represented by the crossings of an equally spaced set of boundaries separated by D. Using quantization, we checks a threshold crossing of output value of a sender whenever an output event occurs and sends the output value to a receiver only when the threshold crossing occurs. The effect of quantization is to reduce the number of messages exchanged between sender and receiver. We can expect to save the communication data and the computation of the receiver from the reduced messages through the message reduction. Considered with scalability of a system, the quantization increases system performance in various ways such as decreasing overall execution time or allowing a larger number of entities to be performed.

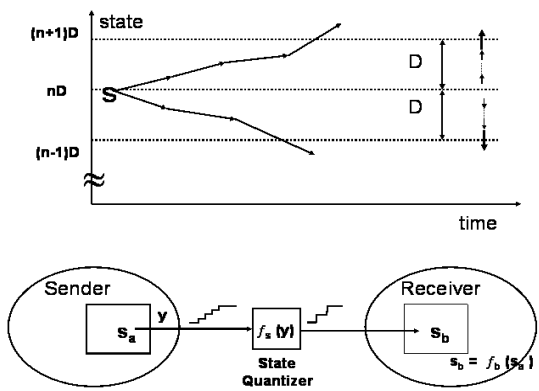


Fig. 1. Quantization and Performance Improvement through Message Traffic Reduction

This paper introduces an actually realized quantization approach to improve system performance through message traffic reduction among system components. As Fig. 1 illustrates, the quantization approach applies when a sender component is updating a receiver component on a numerical, real-valued, state variable, which is a dynamically changing attribute. A quantizer is applied to the sender's output, which checks for threshold (boundary) crossings whenever a change in the variable occurs. Only when such a crossing occurs, a new value of the variable is sent across the network to the receiver. The quantization reduces the number of messages sent and incurs some local computation at the sender.

3 Quantization Component

A quantized integrator [11], which is a quantization component, basically performs a linear extrapolation. The time to the next boundary crossing is the quantum size divided by the input (derivative). The boundary is predicted either to be one up or one down according to the sign of the derivative. When an input event is received, the state is updated using the old input before recalculating the predicted crossing, which provides an important correction for error reduction. A quantized integrator accepts DEVS input segments and produces quantized output.

If we are on a boundary, the time advance computation merely divides quantum interval (D) by the current input x (the derivative or slope). If we reach the upper boundary $(n+1)D$ or lower boundary $(n-1)D$, we output and update the state accordingly. As long as the input remains the same, the time to cross the successive boundaries ($(n+1)D$ or $(n-1)D$) will be the same. When a new input is received, we update the state using the old input and the elapsed time. From this new state (q), the new time to reach either the upper or lower boundary is computed.

Comparison of Time Trajectory of Quantized Integrator
: (Discrete Event System vs. Discrete Time System)

The DEVS stores a state of a system and its last input by definition of DEVS, $M = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta)$. A DTSS system can be strongly represented by the DEVS with four functions, $ta(q,x)$, $\lambda(q,x)$, $\delta_{int}(q,x)$, and $\delta_{ext}((q,x), e, x')$.

The *time advance* is the time to the next system output.

$$ta(q,x) = min(\ t \mid (q, x_{t>})) \neq \phi\}$$

The output of the DEVS at the next internal event is the corresponding system output

$$\lambda(q,x) = \lambda(q, x_{ta(q,x)})$$

Unless there is an external input, the DEVS will update its state to the state of the system at the next output.

$$\delta_{int}(q,x) = \delta_{int}(q, x_{ta(q,x)})$$

If there is an external input after an elapsed time, e, the DEVS will immediately update its state to the corresponding system state and will also store the new input.

$$\delta_{ext}((q,x), e, x') = \delta_{ext}((q, x_{e>}), e, x')$$

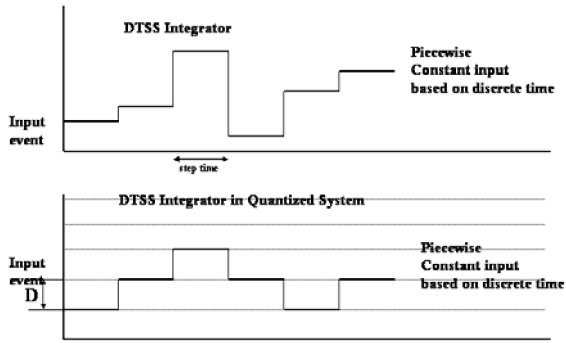


Fig. 2. Input Time Trajectory (DTSS integrator vs. Quantized DTSS Integrator)

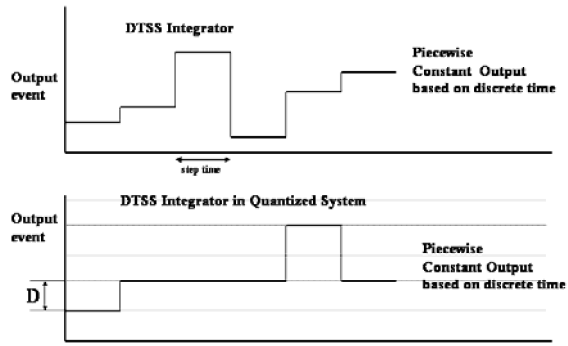


Fig. 3. Output Time Trajectory (DTSS integrator vs. Quantized DTSS Integrator)

Fig. 2 and Fig.3 compare input and output trajectories between of DTSS integrator and Quantized DTSS integrator and show the difference between two integrators that are mentioned with representation of each formalism previously. The integrator designed to fit at DTSS Simulation Environment, after every step time, puts output event and gets input event basically. While the quantized integrator can put output event when output value crosses the boundary of quantum based partition block. And this integrator can get input event when input event occurs for this quantized integrator. Input value is quantized by an input quantizer.

Fig. 4 and Fig. 5 compare input and output trajectories between of DTSS integrator and Quantized DEVS integrator and show the difference between two integrators that are mentioned with representation of each formalism previously. The integrator of DTSS Simulation Environment, after every step time, puts output event and gets input event basically. While the quantized DEVS integrator can put output event at the time from time advance function, *ta()*. Time from time advance function is when state of system crosses the boundary of quantum based partition block of state. That means that the crossing of the partition block boundaries are implemented as state events. So, time from time advance function depends on a quantum, current input, and current state. And this quantized DEVS integrator can get input event when input event occurs for this integrator.

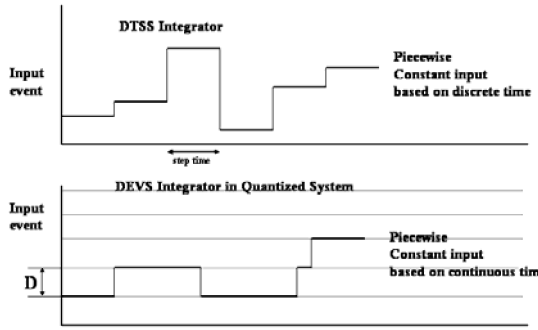


Fig. 4. Input Time Trajectory (DTSS integrator vs. Quantized DEVS Integrator)

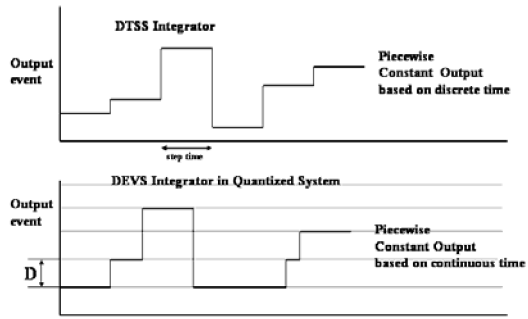


Fig. 5. Output Time Trajectory (DTSS integrator vs. Quantized DEVS Integrator)

4 Kinetics of Spaceship

This section presents a kinetics part of spaceship as an application to evaluate performance of quantized system. We develop a spaceship model as a quantized system. From this model, we construct an abstraction that is for maintaining an accounting of where ships are and predicting their future destinations. Thus our overall modeling objectives are to construct a space travel scheduling and test it. The modeling is based on the differential equations which are based on Newtonian mechanics [13, 14].

4.1 Circulation on an Ideal Circle

Circulation of spaceship on an ideal circle is a part of the kinetics. In order to maintain an ideal circular orbit with radius D and speed v around a massive body, a spaceship is required that a centripetal force, mv^2/d, which equals to the force of gravity. The force of gravity pulls along the line joining the two centers and has magnitude $F = GMm/d^2$, where G is the gravitational constant, M and m are the masses. The distance of a ship with center at (x,y) to the center of gravity of a massive body (x_0,y_0) is $d = ((x - x_0)^2 + (y - y_0)^2)^{1/2}$. The force is projected in the x and y

directions in proportions, $p_x = x/d$ and $p_y = y/d$, respectively. In a ideal orbit with d = D (constant), the coordinate dynamics separate into two independent 2^{nd} order linear oscillators.

Basically, frequency $\omega = (GM/d^3)^{1/2}$ would be to maintain to circulate, however we use a gain value instead of frequency. The gain controls a degree of movement of spaceship. As the gain changes, system performance will be measured and compared since the gain decides stability, accuracy, and execution time. For spaceship traveling, one of the strong influences is gravity. Gravity is a force exerted on a body by all other bodies in relation to their distance away. The center of gravity allows us to aggregate particles in a rigid body into a single point that represents their gravitational interaction with any other body to consider the forces acting at the centers of gravity of interacting bodies.

5 Experiment and Performance Evaluation

We develop a kinetics model of spaceship. The kinetics model has a total of four integrators and is developed on DEVSJAVA modeling and simulation environment [6, 7]. We develop two different systems: Quantized DTSS and Quantized DEVS. Quantized DTSS system includes quantized DTSS integrators and Quantized DEVS system includes quantized DEVS integrators. Fig. 6, Fig. 7, and Fig. 8 compare system accuracy and performance between quantized DTSS and quantized DEVS systems with three performance measures: average of error, number of message passing, and system execution time. As shown in Fig. 6, average of error is increasing apparently as quantum size, D, is increasing. The quantized DEVS system shows the more accuracy than that of the quantized DTSS since DEVS is basically based on a continuous time modeling, thus the error from DEVS is not accumulated. Meanwhile, DEVS is basically based on a discrete time modeling.

The number of message passing and system execution time measure system performance improvement through execution cost reduction. In both of quantized DTSS and quantized DEVS, as quantum size, D, is increasing, number of message passing and system execution time are decreasing apparently in Fig. 7 and Fig. 8. However, the average of error is increasing in Fig. 6. There exists a tradeoff between execution cost reduction and error increment. We should control the quantum size, D, and reduce execution cost within a tolerable error. In comparison between quantized DTSS and quantized DEVS, the quantized DEVS apparently reduces number of message passing and system execution time with the smaller error.

6 Conclusion

This paper presents a quantized system with DTSS and DEVS representations. The quantized system reduces an amount of computation in a complex and large-scale systems, thus reduces a total execution time. Especially, the quantized system apparently reduces the number of massage passing among components. That means that the quantized system reduces data transmission requirement, naturally. The quantized system is able to provide high performance modeling in distributed system

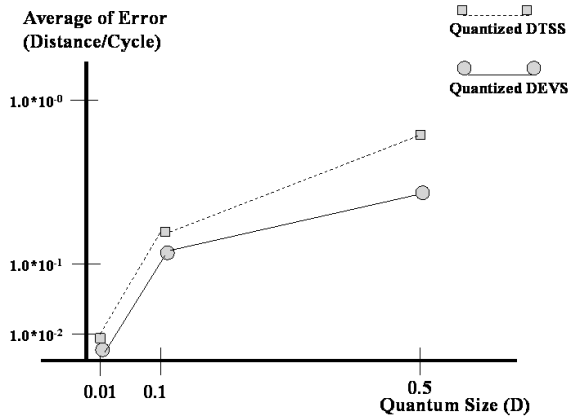


Fig. 6. Average of Error (Quantized DTSS System vs. Quantized DEVS system)

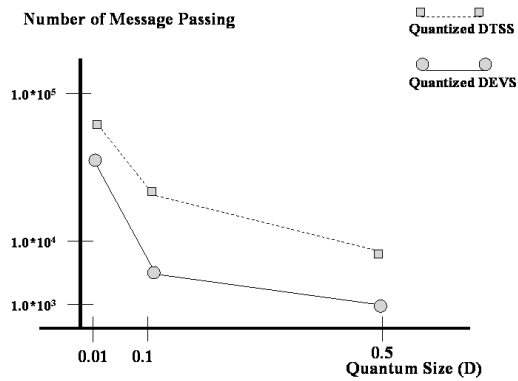


Fig. 7. Number of Message Passing (Quantized DTSS System vs. Quantized DEVS system)

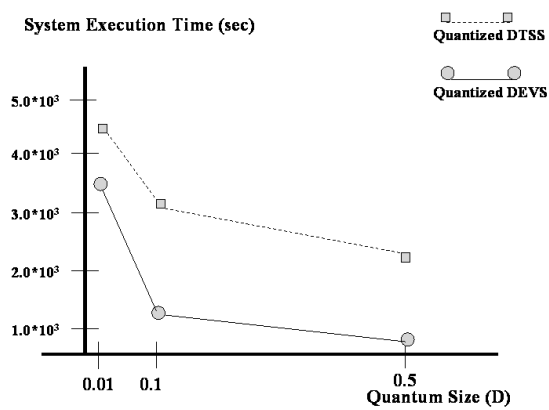


Fig. 8. System Execution Time (Quantized DTSS System vs. Quantized DEVS system)

by reducing data transmission requirement among distributed components. The execution of large-scale distributed system is achieved with high performance modeling in limited communication and computing resources. To realize a quantized system, this paper suggests two types of quantized integrators: quantized DTSS and quantized DEVS. In addition, this paper represents the DTSS formalism in a strong sense of the DEVS, thus this representation shows both continuous and discrete processes can be modeled and executed by the DEVS. Quantized DTSS system is developed with DTSS formalism by using quantized DTSS integrators. Quantized DEVS system is developed with DEVS formalism by using quantized DEVS integrators. The empirical result from the Quantized DTSS and DEVS system shows system performance improvement with a tradeoff from system accuracy. With this limitation, the quantized system should be applied within a tolerable error.

Reference

1. Averill M. Law, and W. David Kelton: Simulation Modeling and Analysis. McGraw-Hill. Inc. (1982)
2. Bernard. P. Zeigler and D. Kim: Design of High Level Modelling / High Performance Simulation Environments. 10th Workshop on Parallel and Distributed Simulation. Philadelphia (1996)
3. Yoonkeon Moon: High Performance Simulation Based Optimization Environment: Modeling Spatially Distributed Large Scale Ecosystems. Ph.D. Dissertation. The University of Arizona (1996)
4. Bernard P. Zeigler, Y. Moon, D. Kim, and J.G. Kim: C++ DEVS: A High Performance Modeling and Simulation Environment. 29th Hawaii International Conference on System Sciences (1996)
5. Zeigler, B.P., et al.: The DEVS Environment for High-Performance Modeling and Simulation. IEEE C S & E (1997) 61–71
6. Zeigler, B.P. and J.S. Lee: Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment. Enabling Technology for Simulation Science(II). SPIE AeoroSense 98. Orlando FL (1998)
7. Zeigler, B.P.: DEVS Theory of Quantization. DARPA Contract N6133997K-0007: ECE Dept. UA Tucson AZ (1998)
8. Ernesto Kofman, Sergio Junco: Quantized-State Systems, a DEVS Approach for Continuous System Simulation. Transactions of SCS (2001)
9. Bernard. P. Zeigler, H. Sarjoughian, and H. Praehofer: Theory of Quantized Systems: DEVS Simulation of Perceiving Agents. J. Sys. & Cyber, Vol. 16, No. 1 (2000)
10. G. Wainer, and B.P. Zeigler: Experimental Results of Timed Cell-DEVS Quantization, AI and Simulation. AIS 2000. Tucson AZ
11. Zeigler, B.P., T.G. Kim, and H. Praehofer: Theory of Modeling and Simulation. 2ed. New York NY: Academic Press (2000)
12. Zeigler, B.P., et al.: DEVS Framework for Modeling, Simulation, Analysis, and Design of Hybrid Systems: Hybrid II, Lecture Notes in CS, P. Antsaklis and A. Nerode, Editors. Springer-Verlag, Berlin (1996) 529–551
13. Roger R. Bate, Donald D. Mueller, Jerry E. White: Fundamentals of Astrodynamics. Dover Publications. New York (1971)
14. Erwin Kreyszig: Advanced Engineering Mathematics**:** Seventh Edition. John Wiley& Sons Inc. New York (1993)

New Digit-Serial Systolic Arrays for Power-Sum and Division Operation in GF(2^m)*

Won-Ho Lee, Keon-Jik Lee, and Kee-Young Yoo

Department of Computer Engineering, Kyungpook National University,
Daegu, 702-701, South Korea
purmi@infosec.knu.ac.kr
kkiir@yahoo.com
yook@knu.ac.kr

Abstract. This paper implements a new digit-serial systolic array for the computation of a power-sum operation and a new digit-serial systolic divider using the proposed systolic power-sum array in GF(2^m) with the standard basis representation. Both of the architectures possess features of regularity, modularity, and unidirectional data flow. As a consequence, they have low $\boldsymbol{AT}$ complexity and are well suited to VLSI implementation with fault-tolerant design. Furthermore, the proposed power-sum array is also possible to select the digit-size of the regular square form.

1 Introduction

The performance of an elliptic curve cryptography (ECC) is primarily determined by the efficient realization of the arithmetic operations in the underlying finite fields GF(2^m). The important operations involved in finite fields GF(2^m) are addition, multiplication, and division. Addition is very simple circuit if the field elements are presented in a polynomial form. However, the other operations are all much more complex. Therefore, coprocessors for ECC are most frequently designed to accelerate the field multiplication and division.

Numerous architectures for the arithmetic operations in GF(2^m) have been reported in previous literatures[1 - 6]. The conventional approaches for computing division in GF(2^m) include the table lookup method, Euclid's algorithm, and Fermat's theorem based method. First, table lookup method is good for small values of m, but its high area complexity makes it difficult for VLSI implementation when m becomes large. Second, the Euclid's algorithm finds the greatest common divisor (GCD) of two polynomials. Although this algorithm can be easily implemented using software, it would be too slow for time critical applications. Finally, Fermat's theorem based method is using successive squaring and multiplication such as $A/B = AB^{-1} = AB^{2^m-2} = A(B(B(B \cdots B(B(B)^2)^2 \cdots)^2)^2)^2$. Therefore, the division and inversion operation can be performed by the iterative application of a power-sum operation $AB^2 + C$.

* This research was supported by University IT Research Center Project.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 638–647, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The bit-parallel systolic architectures for performing the power-sum operation using standard basis representation in GF(2^m) have been proposed [3, 5, 6]. Note that the systolic design in [5] has the bi-directional data flow, while the circuit in [6] has the unidirectional data flow. In this paper, we focus on the digit-serial systolic implementation of power-sum and division operation in GF(2^m) with the standard basis representation.

2 Power-Sum Algorithm in GF(2^m)

Let $A(\alpha)$, $B(\alpha)$, and $C(\alpha)$ be elements in GF(2^m) with a primitive polynomial $G(x)$ of degree m and $G(\alpha) = 0$, where

$$\begin{aligned} A(\alpha) &= \sum_{i=0}^{m-1} a_i \alpha^i = a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} + \cdots + a_1\alpha + a_0, \\ B(\alpha) &= \sum_{i=0}^{m-1} b_i \alpha^i = b_{m-1}\alpha^{m-1} + b_{m-2}\alpha^{m-2} + \cdots + b_1\alpha + b_0, \\ C(\alpha) &= \sum_{i=0}^{m-1} c_i \alpha^i = c_{m-1}\alpha^{m-1} + c_{m-2}\alpha^{m-2} + \cdots + c_1\alpha + c_0, \\ G(\alpha) &\Rightarrow \alpha^m = g_{m-1}\alpha^{m-1} + g_{m-2}\alpha^{m-2} + \cdots + g_1\alpha + g_0. \end{aligned} \tag{1}$$

The coefficients a_i, b_i, c_i, and g_i are the binary digits 0 and 1. As you know, the elements in GF(2^m) can be represented by bit string of length m. For example, $A(\alpha)$ can be represented by bit string $A = (a_{m-1}, a_{m-2}, \ldots, a_1, a_0)$.

Define

$$P(\alpha) = A(\alpha)B^2(\alpha) + C(\alpha) = p_{m-1}\alpha^{m-1} + p_{m-2}\alpha^{m-2} + \cdots + p_1\alpha + p_0. \tag{2}$$

Since $B^2(\alpha) = b_{m-1}\alpha^{2(m-1)} + b_{m-2}\alpha^{2(m-2)} + \cdots + b_1\alpha^2 + b_0 = B(\alpha^2)$, we can derive

$$\begin{aligned} P(\alpha) &= A(\alpha)B(\alpha^2) + C(\alpha) = \sum_{i=0}^{m-1} A(\alpha)b_i\alpha^{2i} + \sum_{i=0}^{m-1} c_i\alpha^i \\ &= \left[\sum_{i=1}^{m-1} A(\alpha)b_i\alpha^{2i} + \sum_{i=2}^{m-1} c_i\alpha^i\right] + A(\alpha)b_0 + c_1\alpha + c_0 \\ &= \left[\sum_{i=1}^{m-1} A(\alpha)b_i\alpha^{2(i-1)} + \sum_{i=2}^{m-1} c_i\alpha^{(i-2)}\right]\alpha^2 + A(\alpha)b_0 + c_1\alpha + c_0 \\ &= \left[\left[\sum_{i=2}^{m-1} A(\alpha)b_i\alpha^{2(i-1)} + \sum_{i=4}^{m-1} c_i\alpha^{(i-2)}\right] + A(\alpha)b_1 + c_3\alpha + c_2\right]\alpha^2 + A(\alpha)b_0 + c_1\alpha + c_0 \\ &= \left[\left[\sum_{i=2}^{m-1} A(\alpha)b_i\alpha^{2(i-2)} + \sum_{i=4}^{m-1} c_i\alpha^{(i-4)}\right]\alpha^2 + A(\alpha)b_1 + c_3\alpha + c_2\right]\alpha^2 + A(\alpha)b_0 + c_1\alpha + c_0 \end{aligned} \tag{3}$$

Further expanding the last summations over i in (3), we obtain the following recursion for $P(\alpha)$:

$$T_i(\alpha) = T_{i-1}(\alpha)\alpha^2 + A(\alpha)b_{m-i} + c_{2(m-i)+1} + c_{2(m-i)}, \qquad (1 \le i \le m) \tag{4}$$

where $T_0(\alpha) = 0$, $P(\alpha) = T_m(\alpha)$, $c_i = 0$ ($m \le i \le 2m$-1), and

$$T_i(\alpha) = t_{i,m-1}\alpha^{m-1} + t_{i,m-2}\alpha^{m-2} + \cdots + t_{i,1}\alpha + t_{i,0}. \tag{5}$$

Substituting (5) into (4) yields

$$\begin{aligned} T_i(\alpha) = {} & t_{i-1,m-1}\alpha^{m+1} + t_{i-1,m-2}\alpha^m + \cdots + t_{i-1,1}\alpha^3 + t_{i-1,0}\alpha^2 \\ & + A(\alpha)b_{m-i} + c_{2(m-i)+1} + c_{2(m-i)} \qquad (1 \le i \le m). \end{aligned} \tag{6}$$

It is also easy to check that

$$\begin{aligned} \alpha^{m+1} &= g_{m-1}\alpha^m + g_{m-2}\alpha^{m-1} + \cdots + g_1\alpha^2 + g_0\alpha \\ &= (g_{m-1}g_{m-1} + g_{m-2})\alpha^{m-1} + (g_{m-1}g_{m-2} + g_{m-3})\alpha^{m-2} + \cdots + g_{m-1}g_0. \end{aligned} \tag{7}$$

Let $\alpha^{m+1} \equiv G'(\alpha) \equiv g'_{m-1}\alpha^{m-1} + g'_{m-2}\alpha^{m-2} + \cdots + g'_1\alpha + g'_0$. Then with (1) and (6), we can rewrite the recursion given in (4) as follows [6]:

$$\begin{aligned} T_i(\alpha) = {} & t_{i-1,m-1}G'(\alpha) + t_{i-1,m-2}G(\alpha) + \cdots + t_{i-1,1}\alpha^3 + t_{i-1,0}\alpha^2 \\ & + A(\alpha)b_{m-i} + c_{2(m-i)+1} + c_{2(m-i)} \qquad (1 \le i \le m) \end{aligned} \tag{8}$$

Based on (8), the power-sum operation can be represented to the bit-wise recurrence equation as following algorithm:

Bit Level Power-Sum Algorithm

Input: $A(\alpha)$, $B(\alpha)$, $C(\alpha)$, $G(\alpha)$

Output: $P(\alpha) = T_m(\alpha) = A(\alpha)B^2(\alpha) + C(\alpha)$

Initial: $(g'_{m-1}, g'_{m-2}, \ldots, g'_0) = (g_{m-1}\cdot g_{m-1} \oplus g_{m-2}, g_{m-1}\cdot g_{m-2} \oplus g_{m-3}, \ldots, g_{m-1}\cdot g_0)$
$(t_{0,m-1}, t_{0,m-2}, \ldots, t_{0,1}, t_{0,0}) = (0,0, \ldots,0,0)$
$(c_{2m-1}, c_{2m-2}, \ldots, c_{m+1}, c_m) = (0,0, \ldots,0,0)$

Recurrence:
1. **for** $i = 1$ **to** m **do**
2. **for** $j = 1$ **to** m **do**
3. **if** $j = m$-1 or m **then** $t_{i-1,\,m-j-2} = c_{2(m-i)+m-j}$
4. $t_{i,m-j} = t_{i-1,m-j-2} \oplus (a_{m-j}\cdot b_{m-i}) \oplus (t_{i-1,m-1}\cdot g'_{m-j}) \oplus (t_{i-1,m-2}\cdot g_{m-j})$

3 Digit-Serial Systolic Array Implementation

3.1 Digit-Serial Systolic Power-Sum Array

The dependence graph (DG) in GF(2^m), obtained from the recurrence equation of above bit level power-sum algorithm is shown in Fig. 1, where $m \times m$ basic nodes are

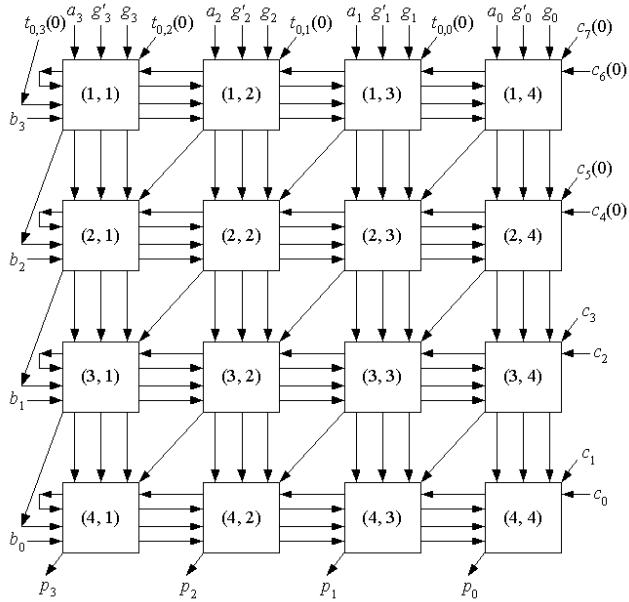


Fig. 1. DG in GF(2^m), where $m = 4$.

used ($m = 4$). In DG, the node means the point at which the computation occurs and the edge means the flow of data.

In Fig. 1, the DG for power-sum operation has a bi-directional data flow in a horizontal direction. As described in [7], a system with unidirectional data flow gains advantages over a system with bi-directional data flow in terms of chip cascade ability, fault tolerance, and possible wafer-scale integration. To overcome such a problem, Wang & Guo [6] combined two adjacent basic cells in the horizontal direction. In other words, to remove a bi-directional data flow, each row of the DG is partitioned into $m/2$ regions by combining only 2 basic cells in a horizontal direction together, and thus, new DG consists of $m \times m/2$ digit cells. However, it is impossible to expand the digit-size for the cell to a $D \times D$ size of the regular square form, instead of 1×2 size. To overcome this problem, the DG of Fig. 1 is reshaped by applying a coordinate transformation to the index space without changing the cell function, i.e., the cell index (i, j) is moved to position $(i, 2i+j-2)$. The resulting DG is shown in Fig. 2 and the circuit of the cell is shown in Fig. 3. It can be seen that this DG involves unidirectional data flow in the horizontal direction, instead of bi-directional data flow.

In Fig. 2, the initial position and data flow of the each variable are as follows: The bit b_{m-i}, $t_{i,m-1}$ and $t_{i,m-2}$ $(0 \le i \le m-1)$, are supplied to point $[i, 0]^T$, and flow into the $[0, 1]^T$ direction without updating. The bit $t_{i,j}$ $(0 \le i \le m-1, -2 \le j \le m-1)$, is supplied to point $[0, j]^T$, and flow into the $[1, 0]^T$ direction. Each bit $t_{i,j}$ is computed at all points represented by rectangles. The bit a_j, g_j, and g'_j $(0 \le j \le m-1)$, are supplied to points $[i, j]^T$, and flow into the $[1, 2]^T$ direction without updating. All computation points on the direction vector $[1, 1]^T$ are executed in parallel. The resulting bits $t_{m,j}$, $(0 \le j \le m-1)$ appear from points $[m, j]^T$.

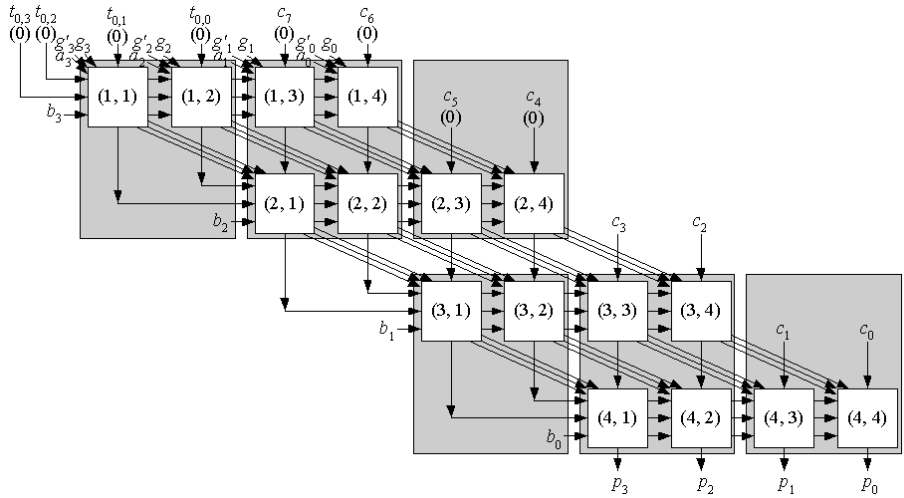


Fig. 2. Modified DG shown in Fig 1.

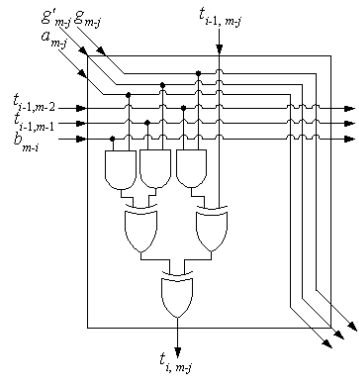


Fig. 3. Circuit of (*i*, *j*) Cell in Fig. 2.

As shown in Fig. 2, if we combine D adjacent basic cells in the horizontal and vertical direction to from a new cell, where D is the digit size. In other words, each row and column of the DG are partitioned into m/D regions by combining only D basic cells together, and thus, a new re-modified DG consists of $m/D \times m/D$ digit cells, where m/D is an integer. In that case, each cell is composed of $3D^2$ 2-input AND gates, $3D^2$ 2-input XOR gates.

By projecting the DG consists of $m/D \times m/D$ digit cells in an east direction following the projection procedure and cut-set systolization [8, 9], a new systolic power-sum array can be easily derived. Fig. 4 shows the digit-serial systolic array for power-sum operation in GF(2^m), where $m = 4$ and $D = 2$. It consists of m/D processing elements (PEs). The square ('■') on the data flow means the buffer for one time step delay. As shown in Fig. 4, since the values broadcasting to all the cells in each row exist, 2-to-1 multiplexers (MUX) and one-bit latches are added for this. These extra

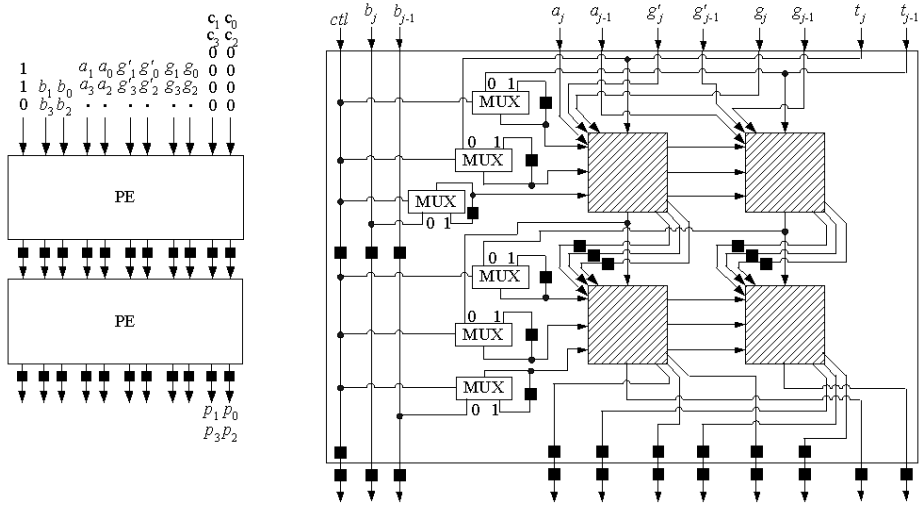


Fig. 4. Digit-serial systolic power-sum array in GF(2^4), and PE structure of array (D = 2).

circuitry operations are controlled by a control signal (*ctl*). The sequence of control signal is 0^1 and $1^{m/2}$, which means one bit of 0 and m/2 bits of 1. The loading operation of the values occurs when the *ctl* is in logic 0.

3.2 Digit-Serial Systolic Divider

Assume that A, B, and D are three elements in GF(2^m), division is performed using multiplication and multiplicative inverse, that is, $D = A/B = AB^{-1}$. Inverse can be regarded as a special case of exponentiation because $B^{-1} = B^{2^m-2} = (B\,(B\,(B \cdots B\,(B\,(B)^2)^2 \cdots)^2)^2)^2$. Therefore, division can be computed as following algorithm:

Division Algorithm

Input: A, B
Output: $D = A/B = AB^{-1}$
Initial: $D = B$
Recurrence: 1. **for** i = 1 **to** m-2 **do**
2. $D = BD^2$
3. $D = AD^2$

Here the result $D = A/B^{-1}$ and the power-sum operations can be used to compute step 2 and step 3 operations. When A = 1, the algorithm realizes the inversion operation B^{-1}.

The above division algorithm can be implemented using digit-serial systolic power-sum array of Fig. 4, as shown in Fig. 5. This array consists of m-1 power-sum arrays for GF(2^m) and some delay elements, where m = 4 and D = 2.

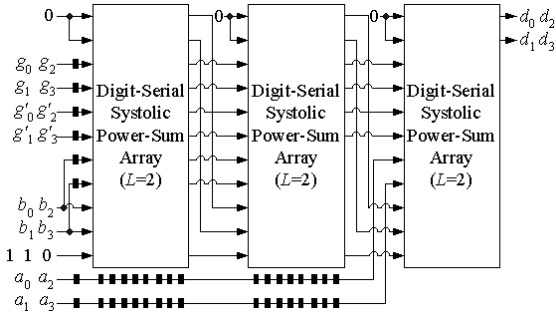


Fig. 5. Digit-serial systolic divider in GF(2^4).

3.3 Analysis

The proposed systolic arrays were described in VHDL with ALTERA MAX PLUS-II tool, and then were simulated using FLEX 10k devices of the ALTERA family for its computation time and correctness.

In order to compare the performance of the proposed systolic arrays with existing architectures, the following assumptions in [10] are made: 1) 3-input and 4-input gate were constructed using two and three 2-input XOR gates, respectively. 2) $T_{XOR2} = 4.2\Delta$, $A_{XOR2} = 14\phi$, $T_{AND2} = 2.4\Delta$, $A_{AND2} = 6\phi$, $T_{MUX2} = 3.8\Delta$, $A_{MUX2} = 14\phi$, $T_L = 1.4\Delta$, $A_L = 8\phi$, where T_{GATE2} and A_{GATE2} are the time and area requirements of a 2-input gate, T_L and A_L are the delay and area of one-bit latch and, Δ and ϕ are the unit gate delay (*ns*) and the number of transistors corresponding to one level of logic circuit, respectively. It show the cost of each gate in terms of the number of transistors it would require when constructed with CMOS technology and the normalized delay of signal propagation through that particular gate.

Comparisons with the characteristics of the systolic architectures described by Wei [5] and Wang & Guo [6] in GF(2^m) are listed in Table 1. In reality, the architectures of [5] and [6] have an I/O format with a bit-parallel-input bit-parallel-output. Whereas, the proposed systolic array has an I/O format with a digit-serial-input digit-serial-output. Table 1 shows the area (***A***) and the computation time (***T***) of one cell (PE) of the proposed systolic array and existing systolic architectures. The ***A***, ***T***, and the area-time (***AT***) complexity of the proposed systolic array for GF(2^m) are as follows:

$$\begin{aligned} &\boldsymbol{A} = (92D^2+106D+8)m/D\phi,\ \boldsymbol{T} = (10.8D+5.2)\Delta \cdot (D+2)m/D \\ &\boldsymbol{AT} = (993.6D^2+3610.4D+3884+1316.8/D+83.2/D^2)\, m^2\phi\Delta \end{aligned} \tag{9}$$

On the other hand, the ***A***, ***T***, and ***AT*** complexity of the systolic arrays of references [5] and [6] are as follows:

$$\begin{aligned} &\boldsymbol{A} = 140\phi \cdot m^2 = 140m^2\phi,\ \boldsymbol{T} = 12.2\Delta \cdot 3m = 36.6m\Delta, \\ &\boldsymbol{AT} = 140m^2\phi \cdot 36.6m\Delta = 5124m^3\phi\Delta \end{aligned} \tag{10}$$

$$\begin{aligned} &\boldsymbol{A} = 256\phi \cdot m^2/2 = 128m^2\phi,\ \boldsymbol{T} = 12.2\Delta \cdot (2m+m/2) = 30.5m\Delta, \\ &\boldsymbol{AT} = 128m^2\phi \cdot 30.5m\Delta = 3904m^3\phi\Delta \end{aligned} \tag{11}$$

Table 1. The comparison of three systolic power-sum arrays in GF(2^m).

	Wei [5]	Wang & Guo [6]	Proposed (Fig. 4)
I/O format	Bit-parallel	Bit-parallel	Digit-serial
Data flow	Bi-directional	Unidirectional	Unidirectional
Number of cells	m^2	$m^2/2$	m/D
Cell complexity	AND2: 3 XOR2: 3 Latch: 10	AND2: 6 XOR2: 6 Latch: 17	AND2: $3D^2$ XOR2: $3D^2$ Latch: $4D^2+8D+1$ MUX2: $3D$
Area per cell	140ϕ	256ϕ	$(92D^2+106D+8)\phi$
Latency	$3m$	$2m+m/2$	$(D+2)m/D$
Critical path	$T_{AND2}+2T_{XOR2}+T_L$	$T_{AND2}+2T_{XOR2}+T_L$	$D(T_{AND2}+2T_{XOR2})+T_{MUX2}+T_L$
Delay per cell	12.2Δ	12.2Δ	$(10.8D+5.2)\Delta$
Control signals	0	0	1

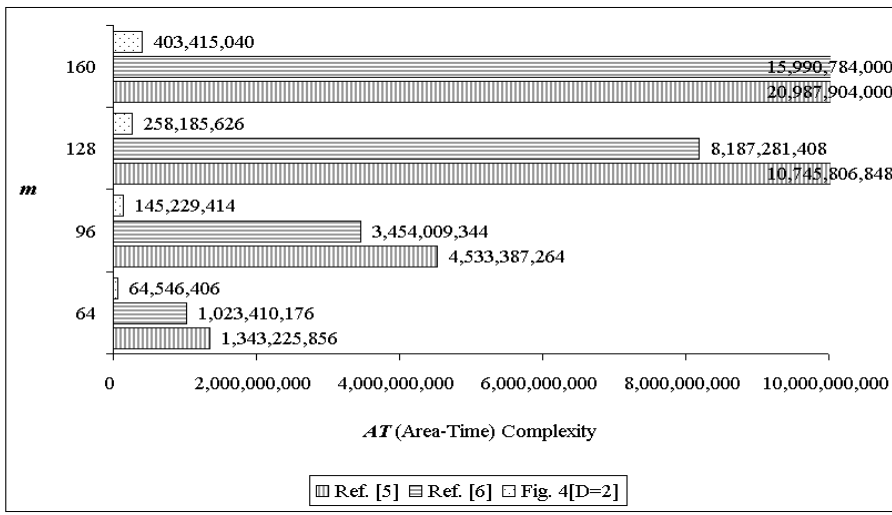


Fig. 6. The ***AT*** complexity of three systolic power-sum arrays in GF(2^m).

The comparison of the ***AT*** complexity of three systolic power-sum arrays in GF(2^m) (for m = 64, 96, 128, 160) is shown in Fig. 6. As can be seen, the proposed systolic array has less the ***AT*** complexity than the existing systolic arrays.

Fig. 7 shows the comparison of the ***AT*** complexity of the proposed power-sum array in GF(2^{160}) using the various digit size D (for D = 2, 4, 5, 8, 10, 16, 20, 32, 40, 80). As shown in Fig. 7, the proposed systolic array has much smaller ***AT*** complexity when D = 2.

Therefore, the proposed systolic power-sum array has less ***AT*** complexity than the existing systolic arrays and it has much smaller ***AT*** complexity when D = 2, although it has one control signal.

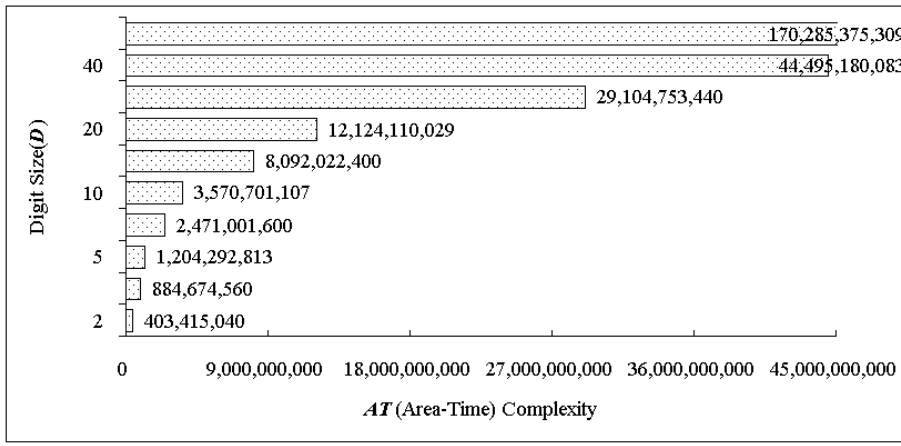


Fig. 7. The ***AT*** complexity of the proposed power-sum array in GF(2^{160}).

Table 2. Comparison of three systolic arrays for division in GF(2^m).

		Wei [5]	Wang & Guo [6]	Proposed (Fig. 5)
I/O format		Bit-parallel	Bit-parallel	Digit-serial
Data flow		Bi-directional	Unidirectional	Unidirectional
Number of cells		$m^2(m\text{-}1)$	$m^2(m\text{-}1)/2$	$m(m\text{-}1)/D$
Total complexity	AND2	$3m^2(m\text{-}1)$	$3m^2(m\text{-}1)$	$3m(m\text{-}1)D$
	XOR2	$3m^2(m\text{-}1)$	$3m^2(m\text{-}1)$	$3m(m\text{-}1)D$
	Latch	$16m^3\text{-}20m^2$	$(21m^3\text{-}12m^2\text{-}7m)/2$	$m(m\text{-}1)(4D\text{+}8\text{+}1/D)$
	MUX2	0	0	$3m(m\text{-}1)$
Latency		$2m^2\text{-}m$	$2m^2\text{-}3m/2$	$(m^2(D\text{+}1)\text{+}m)/D$
Critical path		$T_{AND2}\text{+}2T_{XOR2}\text{+}T_L$	$T_{AND2}\text{+}2T_{XOR2}\text{+}T_L$	$D(T_{AND2}\text{+}2T_{XOR2})\text{+}T_{MUX2}\text{+}T_L$
AT complexity		$\mathbf{O}(m^5)$	$\mathbf{O}(m^5)$	$\mathbf{O}(m^4)$
Control signals		0	0	1

Table 2 gives some comparisons of the proposed digit-serial systolic divider with the related systolic dividers described in [5] and [6]. As shown in Table 2, the proposed systolic divider has less ***AT*** complexity than the existing systolic arrays for division in GF(2^m), although it has one control signal.

4 Conclusion

In this paper, we have presented two digit-serial systolic arrays for performing power-sum and division operation in GF(2^m) with the standard basis representation. The proposed systolic arrays have a significant improvement in reducing the ***AT*** complexity compared with previous architectures. In particular, the proposed power-

sum array has much smaller $\boldsymbol{AT}$ complexity when $D = 2$. Furthermore, it is also possible to select the digit-size of the regular square form. Both of the architectures possess features of regularity, modularity, and unidirectional data flow. Thus, they are well suited to be implemented using VLSI techniques.

References

1. C. L. Wang and J. L. Lin: Systolic array implementation of multipliers for finite field GF(2^m). IEEE Trans. Circuits System. Vol. 38. (1991) 796–800
2. C. L. Wang and J. L. Lin: A systolic architecture for inverses and divisions in GF(2^m). IEEE Trans. Computer. Vol. 42. (1993) 1141–1146
3. S. W. Wei: A systolic power-sum for GF(2^m). IEEE Trans. Computer. Vol. 43. (1994) 226–229
4. J. H. Guo and C. L. Wang: Bit-serial Systolic Array Implementation of Euclid's Algorithm for Inversion and Division in GF(2^m). Proc. 1995 Int. Symp. VLSI Technology, Systems, and Applications. (1997) 113–117
5. S. W. Wei: VLSI architectures for computing exponentiation, multiplicative inverses, and divisions in GF(2^m). Proc. 1995 IEEE Int. Symp. Circuits and System. (1995) 4.203–4.206
6. C. L. Wang and J. H. Guo: New Systolic Array for $C + AB^2$, Inversion and Division in GF(2^m), IEEE Trans. Computer. Vol. 49. (2000) 1120–1125
7. J. V. McCanny, R. A. Evans, and J. G. Mcwhirter: Use of unidirectional data flow in bit-level systolic array chips, Electron. Letters. Vol. 22. (1986) 540–541
8. S. Y. Kung: VLSI array processors. Prentice Hall, New Jersey. (1988)
9. K. Y. Yoo: A Systolic Array Design Methodology for Sequential Loop Algorithms. Ph.D. thesis, Rensselaer Polytechnic Institute. (1992)
10. Daniel. D. Gajski: Principles of Digital Design. Prentice Hall, Upper Saddle River, New Jersey. (1997)

Generation of Unordered Binary Trees

Brice Effantin*

Istituto di Informatica e Telematica, CNR,
Area della Ricerca, Via Moruzzi 1, 56124 PISA, ITALY
brice.effantin@iit.cnr.it

Abstract. A binary unordered tree is a tree where each internal node has two children and the relative order of the subtrees of a node is not important (i.e. two trees are different if they differ only in the respective ordering of subtrees of nodes). We present a new method to generate all binary rooted unordered trees with n internal nodes, without duplications, in $O(\log n)$ time.

1 Introduction

The production of lists of all the combinatorial configurations of a certain type is often useful. The generation of such lists of all shapes of specified kind is therefore a matter of some interest. Thus we are interested on the generation of unordered rooted trees. Generating these trees is necessary in both the graph theory and in different applications. For example, a list of all trees with a given number of internal nodes can be used in computer science to test or analyze an algorithm for its correctness or computational complexity.

They are many special types of rooted trees. If the relative order of the subtrees of a node is important, we say the tree is an *ordered* tree. If we do not care to regard two trees as different when they differ only in the respective ordering of subtrees of nodes, the tree is said to be *unordered*. Actually at each unordered tree implicitly corresponds many ordered trees obtained by commutation of all internal node subtrees.

The generation of combinatorial objects is a well-known problem. In [5], McKay describes a general technique for generating families of combinatorial objects without isomorphs. Moreover, many algorithms has been developed to generate rooted trees and free trees. In [1], Beyer and Hedetniemi give a constant time algorithm to generate rooted trees. This algorithm uses level sequences and generates these sequences in reverse lexicographic order. In [2], Kozina gives another method for coding and generating rooted trees which has running time of $O(nt)$, where t is the number of rooted trees on n nodes. Wright and *al.*, in [9], extend an algorithm of Beyer and Hedetniemi to generate unlabeled free trees. They show that their algorithm only requires $O(n)$ space and a constant average time per tree independent of n. In 1986, Pallo introduced a new coding in [7], and in [8], he gives an algorithm to generate lexicographically the binary unordered trees. He shows experimentally that his algorithm is in constant amortized time per tree. Kubicka and Kubicki also give in [3], a similar approach as Beyer and

* Brice Effantin acknowledges the finantial support provided through the European Community's Human Potential Program under the contract HPRN-CT-2002-00278, COMBSTRU.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 648–655, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Hedetniemi for generating binary rooted trees. They prove that the average number of steps their algorithm has to perform per tree is bounded by a constant independent of the order of the trees. More recently, Li and Ruskey [4] present a new approach to the exhaustive generation of rooted and free trees. Their algorithms use a linear space and have running times that are proportional to the number of trees produced.

In this paper we present a new recursive algorithm for generating binary unordered trees with n internal nodes. Generally speaking, recursive algorithms are simpler and allow easily to manipulate the subtrees. In section 2, we present the coding used for the trees and we define a canonical representation of the unordered trees. Next in section 3, we exhibe an efficient algorithm for generating systematically all these representations, and we give an analysis of this algorithm where we show that each tree is generated in a $O(\log n)$ time.

2 Preliminaries

In a binary tree, every node, except the root, has a parent. Each internal node has a left and a right son. External nodes, called *leaves*, have no children. Let T be a binary tree on n internal nodes. We denote by T_L and T_R the left and right subtrees of T. The level of a node x of T, denoted $L(x)$, is one more the distance between the root and x. The level of the root is $L(root) = 1$. The weight $|T|$ of a tree T is defined as the number of external nodes of T, and the height of T is $h(T) = \max(L(x) - L(r)|x$ is a leaf of T and r is the root of $T)$. The Catalan number $B_n = \frac{(2n)!}{n!(n+1)!}$ gives the number of ordered trees with n internal nodes ($n + 1$ leaves).

The *weight coding* of a tree T on n internal nodes, denoted by $W(T)$, is the $2n$ integers sequence given by $W(T) = (|T_L|, W(T_L), |T_R|, W(T_R))$, where $W(T_L)$ (resp. $W(T_R)$) is computed if T_L (resp. T_R) admits at least one internal node. Figure 1 shows some weigh codings of ordered trees $\in B_6$. Then each node of the tree (except the root) is represented in the weight coding: the value 1 represents a leaf and the other values represent internal nodes. In this paper, an integer i in a weight coding, with $1 \leq i \leq 2n$, representing the number of leaves in the left (or right) subtree, will be assimilate to the root of this subtree (for example, in the first weight coding in Figure 1, *i.e.* 542112111211,

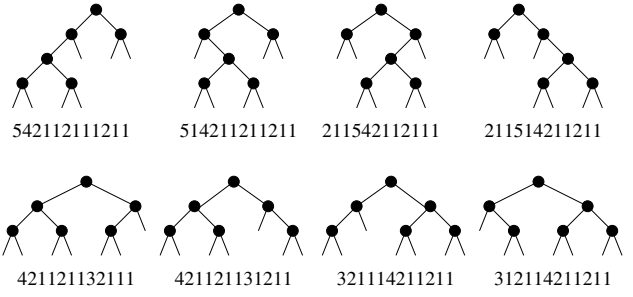


Fig. 1. The trees of the first (resp. second) line will be considered to be different as ordered trees, although they would be the same as unordered trees.

Table 1. The numbers of ordered and unordered trees having fewer than twelve internal nodes

n	B_n	c_n
3	5	2
4	14	3
5	42	6
6	132	11
7	429	23
8	1430	46
9	4832	98
10	16796	207
11	58786	451
12	208012	983

the two integers $i = 1$ and $j = 10$ represent respectively the left and the right son of the root of the tree).

Let C_n be the set of unordered trees with n internal nodes. Let c_n be the number of elements of C_n. Then, c_n can be computed by the recurrence formulas [6]:

$$\begin{cases} c_0 = c_1 = 1, \\ c_{2k} = c_0 c_{2k-1} + c_1 c_{2k-2} + \ldots + c_{k-1} c_k, \\ c_{2k+1} = c_0 c_{2k} + c_1 c_{2k-1} + \ldots + c_{k-1} c_{k+1} + \frac{1}{2} c_k (c_k + 1) \end{cases}$$

The Table 1 shows the numbers of ordered and unordered trees having fewer than twelve internal nodes.

The eight ordered trees in Figure 1 are different, but there are only two unordered trees (one on each line). We give now a *canonical weight coding* to get a unique representation of a tree T. Given an unordered tree T, the canonical weight coding of T is the weight coding verifying, for any subtree t of T:

- $|t_L| > |t_R|$ or
- $|t_L| = |t_R|$ and $h(t_L) \geq h(t_R)$.

For example, the canonical weight codings of the two unordered trees of Figure 1 are (542112111211) for the first line and (421121132111) for the second line.

3 Generating Algorithm

This algorithm computes the next canonical weight coding from the previous. From the end of a weight coding, we find the integer i such that $i \geq 3$, and i represents an internal node belonging to a left subtree t_L of at least one subtree t of T. Then we decrease this integer and the weight coding is completed.

We present two procedures, $Generate$ and $PutTree$, which will effectively generate all canonical weight codings of unordered trees in C_n. We define two global variables used by the two procedures:

- Let n be the number of internal nodes
- Let `W` be a table of size $2n + 1$ (0 to $2n$) containing the current weight coding

In the procedure $Generate$, we find the next integer i which will be decreased, with $1 \leq i \leq 2n$. Then the procedure $PutTree$ computes the new value of each integer of the subtree rooted in i. Next, the procedure $Generate$ completes the weight sequence.

3.1 Procedure *Generate*

This procedure is the main procedure. It computes the search of the next integer which will be decreased and completes the weight sequence.

Procedure 1 *: Generate()*

```
BEGIN

(01) for i:=0 to n-1 do
(02)   W[i]:=n-i+1
(03) enddo
(04) for i:=n to 2n do
(05)   W[i]:=1
(06) enddo
(07) Output()
(08) Next:=1
(09) while(Next≠0) do
(10)   Next:=max{i|0≤i≤2n-3, W[i-1]>W[i] and W[i]>⌊(W[i-1]+1)/2⌋}
(11)   if Next≠0 then
(12)     PutTree(Next,W[Next]-1,Next-1)
(13)     Son:=Next-1
(14)     while(Son>0) do
(15)       if W[Son-1]>W[Son] then
(16)         Son:=Son-1
(17)         if 2.(W[Son]-W[Son+1])=W[Son] then
(18)           for i:=Son+1 to Son+2.W[Son+1]-1 do
(19)             W[Son+2.W[Son+1]+i-(Son+1)]:=W[i]
(20)           enddo
(21)         else
(22)            PutTree(Son+2.W[Son+1], W[Son]-W[Son+1],Son)
(23)         endif
(24)       else
(25)         Son:=max{i|0≤i≤Son, Son=i+2.W[i+1]
             and W[Son]=W[i]-W[i+1]}
(26)       endif
(27)     enddo
(28)     Output()
(29)   endif
(30) enddo

END
```

We find some parts in this procedure. First, the lines (01) to (06) initialize the table W and give the first canonical weight coding (each weight coding is given by integers 1 to $2n$ of W).

In line (10), Next contains the next integer which will be computed. To avoid generating duplications and as in a canonical weight coding for each subtree t of T we have $h(t_L) \geq h(t_R)$, Next must be the left son of the root of T', where T' is a subtree of T. So we must have W[Next-1]>W[Next]. Moreover, for a canonical weight coding, we must have in each subtree t of T more or equal nodes in t_L than in t_R. Therefore, $\texttt{W[Next]} > \left\lfloor \frac{\texttt{W[Next-1]+1}}{2} \right\rfloor$.

The new value of the integer Next (and integers corresponding to the nodes of its subtree) is given by line (12).

The lines (13) to (27) show how we complete the weight coding. Let t be the subtree rooted in (Son-1). In the *while* loop (lines (14) to (27)), we complete the right subtree of t if Son is the left son of (Son-1) (i.e. W[Son-1]>W[Son]). This step is computed for each subtree containing Son and rooted in levels $L(\texttt{Son})$ to 1.

In lines (17) to (23), we separate two cases to avoid having a right subtree higher than the left and so to avoid duplications.

In the line (25), Son was not the left son of the subtree rooted in (Son-1) and we search the father of Son.

3.2 Procedure *PutTree*

This procedure put a value Val on an integer Son of the weight coding, and recursively, it completes the subtree rooted in Son.

Procedure 2 *: PutTree(Son,Val,Father)*

```
BEGIN

(01)  W[Son]:=Val
(02)  if W[Son]>1 then
(03)    PutTree(Son+1,Val-1,Son)
(04)  endif
(05)  if Son=Father+1 then
(06)    if 2.(W[Father]-W[Son])=W[Father] then
(07)      for i:=Son to Father+2.W[Son]-1 do
(08)        W[Father+2.W[Son]+i-Son]:=W[i]
(09)      enddo
(10)    else
(11)       PutTree(Father+2.W[Son], W[Father]-W[Son],Father)
(12)    endif
(13)  endif

END
```

In this procedure, we find two parts. Firstly, in line (01) we put a new value on the integer Son. Secondly, in the lines (02) to (13), we complete the subtree rooted on Son. Let t be the subtree rooted on Son and t' the subtree rooted on Father. The lines

(02) to (04) complete the left subtree of t. The line (05) determines if Son is the left son of Father. Then lines (06) to (12) compute the integers corresponding to t'_R, as in lines (17) to (23) in Procedure 1.

Table 2. List of the canonical weight sequences of the 46 unordered trees of C_8

8765432111111111	8632111321112111	7432111132111211
8765421121111111	8543211111321111	7421121132111211
8765321112111111	8542112111321111	6543211111132111
8764321111211111	8532111211321111	6542112111132111
8764211211211111	8432111143211111	6532111211132111
8763211132111111	8432111142112111	6432111121132111
8754321111121111	8421121142112111	6421121121132111
8754211211121111	7654321111111211	6321113211132111
8753211121121111	7654211211111211	5432111114321111
8743211113211111	7653211121111211	5432111114211211
8742112113211111	7643211112111211	5421121114321111
8654321111112111	7642112112111211	5421121114211211
8654211211112111	7632111321111211	5321112114321111
8653211121112111	7543211111211211	5321112114211211
8643211112112111	7542112111211211	
8642112112112111	7532111211211211	

Table 3. Number of integers modified during the execution of the algorithm.

n	c_n	Total number of integers modified	Number of integers modified (per tree)
3	2	4	2
4	3	6	2
5	6	19	3,167
6	11	36	3,273
7	23	95	4,130
8	46	191	4,152
9	98	450	4,592
10	207	961	4,643
11	451	2223	4,929
12	983	4877	4,961
13	2179	11115	5,101
14	4850	24844	5,122
15	10905	56871	5,215
16	24631	128822	5,230
17	56011	295260	5,271
18	127912	675505	5,281
19	293547	1558213	5,308
20	676157	3594053	5,315

3.3 Analysis of the Algorithm

Firstly, we give in Table 2 the list of the canonical weight codings of the unordered trees of C_8 computed with our algorithm.

Since we store only the current weight coding, the space complexity of this algorithm is $O(n)$. For the time complexity, one can see that the lines `(10)` and `(25)` (Procedure 1), where we search the positions of some nodes, downgrade the efficiency of the algorithm. Indeed, the search of the next integer to decrease (line `(10)`) can traverse all the weight coding. Then, this line is computed in $O(n)$. By the same way, the line `(25)` is also computed in $O(n)$. The efficiency of these two parts can be very upgraded by adding two tables. Let P and F be respectively a table of up to n integers used to pile up the successive values of `Next`, and a table of $2n + 1$ integers used to store the father of each integer. In this case, the lines (10) and (25) are computed in $O(1)$ and the space complexity is always in $O(n)$.

Thus, the time complexity of the algorithm is proportional to the number of integers which will be modified for each tree. Then, we modify at most all the integers of the coding and therefore the time complexity is $O(n)$. But, in a lot of cases, only a part of the coding is traversed. Indeed, a large number of calculations affects only the right subtree of some subtrees of T. So, the integers corresponding to some left subtrees will not be modified. Thus, we observe experimentally the results given in the Table 3. From these results, we deduce that the time complexity is $O(\log n)$ per tree. This is explained by the evolution of c_n (indeed, c_n has the same evolution as B_n whose asymptote is 4^n).

4 Conclusion

The interest of determining several methods for problems already studied can be found in the solving of larger problems. Indeed, suppose that a problem P (the generation of unordered binary trees for example) is a step in the solving of a larger problem P'. Then, the method choosen for the problem P will be the most interesting for the problem P'. Thus in some cases, a $O(\log n)$ method can be more interesting than a $O(1)$ method plus an $O(n)$ algorithm to transform the tree in the used coding.

References

1. Beyer, T., Hedetniemi, M.: Constant Time Generation of Rooted Trees. SIAM Journal on Computing **9** (1980) 706–712
2. Kozina, A.V.: Coding and Generation of Nonisomorphic Trees. Cybernetics **25** (1979) 645–651
3. Kubicka, E., Kubicki, G.: Constant Time Algorithm for Generating Binary Rooted Trees. Congressus Numerantium **90** (1992) 57–64
4. Li, G., Ruskey, F.: The Advantages of Forward Thinking in Generating Rooted and Free Trees. Extended Abstract SODA 1999
5. McKay, B.D.: Isomorph-Free Exhaustive Generation. Journal of Algorithms **26** (1998) 306–324
6. Murtagh, F.: Counting dendrograms: a survey. Discrete Applied Mathematics **7** (1984) 191–199
7. Pallo, J.M.: Enumerating, Ranking and Unranking Binary Trees. The Computer Journal **29** (1986) 171–175

8. Pallo, J.M.: Lexicographic Generation of Binary Unordered Trees. Pattern Recognition Letters **10** (1989) 217–221
9. Wright, R.A., Richmond, B., Odlyzko, A., McKay, B.D.: Constant Time Generation of Free Trees. SIAM Journal on Computing **15** (1986) 540–548

A New Systolic Array for Least Significant Digit First Multiplication in $GF(2^m)$

Chang Hoon Kim[1], Soonhak Kwon[2], Chun Pyo Hong[3], and Hiecheol Kim[3]

[1] Dept. of Computer and Information Engineering, Daegu University, Jinryang, Kyungsan, 712-714, Korea
[2] Dept of Mathematics and Institute of Basic Science, Sungkyunkwan University, Suwon, 440-746, Korea
[3] Dept. of Computer and Communication Engineering, and Institute of Ubiquitous Computing, Daegu University, Jinryang, Kyungsan, 712-714, Korea
chkim@dsp.taegu.ac.kr, shkwon@math.skku.ac.kr

Abstract. This paper presents a new digit-serial systolic multiplier over $GF(2^m)$ for cryptographic applications. When input data come in continuously, the proposed array produces multiplication results at a rate of one every $\lceil m/D \rceil + 2$ clock cycles, where D is the selected digit size. Since the inner structure of the proposed array is tree-type, critical path increases logarithmically proportional to D. Therefore, the computation delay of the proposed architecture is significantly less than previously proposed digit-serial systolic multipliers whose critical path increases proportional to D. Furthermore, since the new architecture has the features of regularity, modularity, and unidirectional data flow, it is well suited to VLSI implementations.

Keywords: Cryptography, Finite Field Multiplication, Digit-Serial Architecture, Systolic Array, VLSI.

1 Introduction

In recent years, finite field $GF(2^m)$ has been widely used in various applications such as error-correcting code and cryptography [1-2]. Important operations in $GF(2^m)$ are addition, multiplication, exponentiation, and division. Since addition in $GF(2^m)$ is bit independent XOR operation, it can be implemented in fast and inexpensive ways. The other operations are much more complex and expensive. This paper focuses on the hardware implementation of fast and low-complexity digit-serial multiplier over $GF(2^m)$, since computing exponentiation and division can be performed by repeated multiplications.

Many approaches and architectures have been proposed to perform $GF(2^m)$ multiplication [3-14]. The most commonly used basis representations are dual, normal, and standard basis. Multipliers using the dual and normal basis representations require a basis conversion, in which complexity heavily depends on the irreducible polynomial $G(x)$. In contrast, multipliers that use the standard basis do not require a basis conversion; they are therefore more efficient from the point of view of irreducible polynomial selection and hardware optimization [4].

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 656–666, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Recently, Song et al. [11] proposed fast and low-energy digit-serial/parallel multipliers for $GF(2^m)$ using special irreducible polynomials (a polynomial of the form $G(x) = x^m + g_k x^k + \sum_{i=0}^{k-1} g_i x^i$, where $D \leq m-k-1$). If this irreducible polynomial is used, $A(x)x^D \mod G(x)$ operation can be computed using bit-wise AND gates and binary tree of XOR gates [11], where $A(x)$ is an element in $GF(2^m)$. Unlike other special irreducible polynomials such as trinomials and all one polynomials, we can easily find, for any m, such irreducible polynomials by appropriately selecting the digit size D. As presented in [16], for elliptic curve cryptosystems (ECC), we can select various D. One can find field size and corresponding irreducible polynomials, for ECC, in [16].

Although the multipliers proposed by Song et al. have many advantages in terms of computation delay, energy, and irreducible polynomial selection, they are not systolic architecture. In other words, they include many global signals broadcasting. Accordingly, if m gets large, the signal propagation delay also increases. This is a great drawback for cryptographic applications.

In this paper, we propose a new digit-serial systolic multiplier over $GF(2^m)$ for cryptographic applications. From the least significant digit first (LSD-first) multiplication algorithm [11], we obtain new dependence graphs (DGs) of digit level and design a new digit-serial systolic array based on the new DGs. When input data come in continuously, the proposed array produces multiplication results at a rate of one every $N+2$ clock cycles. Since the inner structure of the proposed array is tree-type, critical path increases logarithmically proportional to D. Therefore, the computation delay of the proposed architecture is significantly less than previously proposed digit-serial systolic multipliers whose critical path increases proportional to D. Furthermore, since the proposed architecture has the features of regularity, modularity, and unidirectional data flow, it is well suited to VLSI implementations.

2 LSD-First Multiplication Algorithm in $GF(2^m)$

Let $A(x) = \sum_{i=0}^{m-1} a_i x^i$ and $B(x) = \sum_{i=0}^{m-1} b_i x^i$ be two elements in $GF(2^m)$, $G(x) = x^m + \sum_{i=0}^{m-1} g_i x^i$ be the irreducible polynomial used to generate the field $GF(2^m) \cong GF(2)[x]/G(x)$, and $P(x) = \sum_{i=0}^{m-1} p_i x^i$ be the result of the multiplication $A(x)B(x) \mod G(x)$. Let digit-size=D and N denote the total number of digits with $N = \lceil m/D \rceil$. We define the digit A_i $(0 \leq i \leq N-1)$ as following.

$$A_i = \begin{cases} \sum_{j=0}^{D-1} a_{Di+j} x^j, & 0 \leq i \leq N-2 \\ \sum_{j=0}^{m-1-D(N-1)} a_{Di+j} x^j, & i = N-1 \end{cases} \tag{1}$$

The digits B_i, G_i, and P_i are defined similarly. To compute the multiplication $A(x)B(x) \mod G(x)$, we can use the following LSD-first scheme.

$$\begin{aligned} P(x) &= A(x)B(x) \mod G(x) \\ &= B_0A(x) + B_1[A(x)x^D \mod G(x)] + B_2[A(x)x^{2D} \mod G(x)] \\ &\quad + \cdots + B_{N-1}[A(x)x^{D(N-1)} \mod G(x)] \end{aligned} \tag{2}$$

Based on (2), we can derive the following LSD-first multiplication algorithm [11].

[Algorithm I] LSD-First Multiplication Algorithm in $GF(2^m)$ [11]
Input: $G(x), A(x), B(x)$
Output: P has $P(x) = A(x)B(x) \mod G(x)$
Initialize: $A = A^{(0)} = A(x)$, $B = B(x)$, $G = G(x)$, $P = P^{(0)} = 0$
1. **for** $i = 1$ **to** N **do**
2. $A^{(i)} = A^{(i-1)}x^D \mod G$
3. $P^{(i)} = B_{i-1}A^{(i-1)} + P^{(i-1)}$
4. **end for**
5. $P = P^{(N)} \mod G$

3 New Dependence Graphs for LSD-First Multiplication in $GF(2^m)$

3.1 Main Operations for the Algorithm I

Before implementing the LSD-first multiplication algorithm, we consider its main operations. From Algorithm I, we can notice that the operation $A^{(i)} = A^{(i-1)}x^D \mod G$ and $P = P^{(N)} \mod G$ are exactly the same. Therefore, we will consider $A^{(i)} = A^{(i-1)}x^D \mod G$ and $P^{(i)} = B_{i-1}A^{(i-1)} + P^{(i-1)}$ operations. if $D \leq (m-k-1)$, the operation $A^{(i)} = A^{(i-1)}x^D \mod G$ can be computed using bit-wise AND gates and binary tree of XOR gates, where k is the second highest degree of $G(x)$ [11]. We describe an example in Fig. 1, where $m = 9$ and $D = 3$.

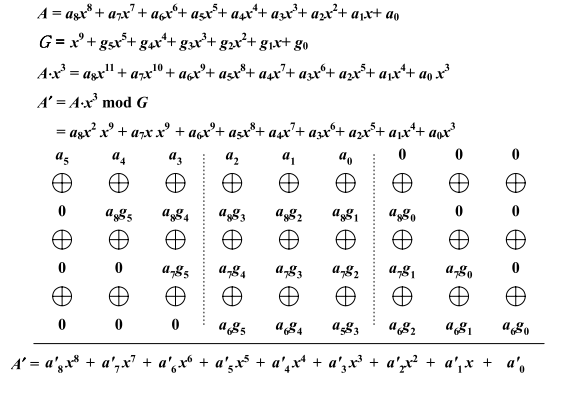


Fig. 1. An example for $A(x)x^D \mod G(x)$ with $m = 9$ and $D = 3$

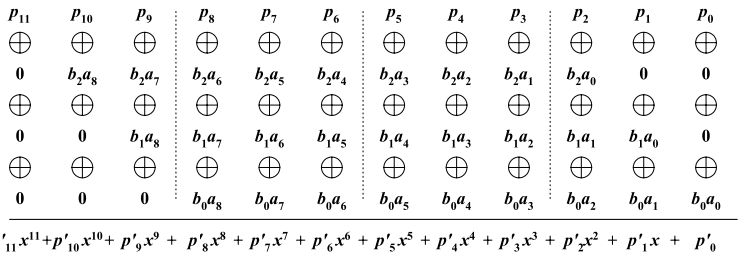


Fig. 2. An example for $B_iA + P$ with $m = 9$ and $D = 3$

Let $A' = A \cdot x^D \mod G$. We can compute the coefficients A'_i using the following equation (3).

$$A'_i = \sum_{i=0}^{N-1} \sum_{j=0}^{D-1} (\sum_{k=1}^{D} (a_{m-k} g_{D(i-1)+j+k}) + a_{D(i-1)+j}) \tag{3}$$

where $A_{-1} = G_{N-1} = G_{-1} = 0$.

In addition, we show an example of $P' = B_iA + P$ operation in Fig. 2, where $m = 9$ and $D = 3$. As described in Fig. 2, although the degree of P is $(m+D-2)$ at most, we add 1 bit redundancy to obtain the same basic cell of digit-level. Similar to (3), we have the following equation (4).

$$P'_i = \sum_{i=0}^{N} \sum_{j=0}^{D-1} (\sum_{k=1}^{D} (b_{D-k} a_{D(i-1)+j+k}) + p_{Di+j}) \tag{4}$$

where $A_{-1} = G_{N-1} = G_{-1} = 0$.

3.2 New Dependence Graphs for Digit-Serial Multiplication in $GF(2^m)$

Based on (3), we can derive a new DG for $A^{(i-1)}x^D \mod G$ operation as shown in Fig. 3. From the Algorithm I, since we do not need to compute the final iteration for the result of $A^{(N-1)}x^D \mod G$, the DG corresponding to the operation consists of $(N-1) \times N$ basic cells of digit-level. In particular, we assumed $m = 9$ and $D = 3$ in the DG of Fig. 3 and Fig. 4 represents the architecture of basic cell. The cells in the i-th row of the array perform the i-th iteration of $A^{(i-1)}x^D \mod G$ operation. The coefficients of the intermediate result A^i emerge from each bottom row of the array.

Fig. 5 represents a new DG for $P^{(i)} = B_{i-1}A^{(i-1)} + P^{(i-1)}$ and $P = P^{(N)} \mod G$ (step 5 of the Algorithm I) operations. The DG consists of $(N+1) \times N$ Type-1 cells and N Type-2 cells of digit-level. In Fig. 5, we assumed $m = 9$ and $D = 3$. The Type-1 cells in the i-th row of the array compute $P^{(i)} = B_{i-1}A^{(i-1)} + P^{(i-1)}$ and the Type-2 cells in the $(N+1)$-th row perform $P = P^{(N)} \mod G$ operation respectively. The structure of Type-1 cell is shown in Fig. 6. Since

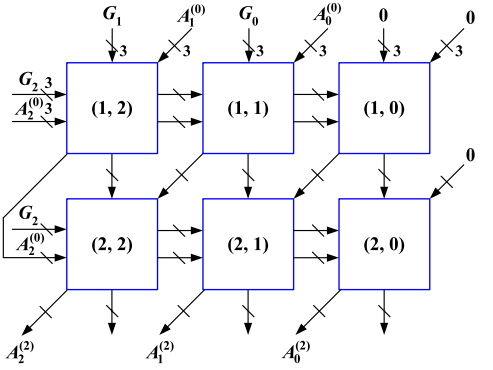


Fig. 3. Digit-level DG for $A^{(i-1)}x^D \mod G$ operation in $GF(2^9)$ with $D = 3$

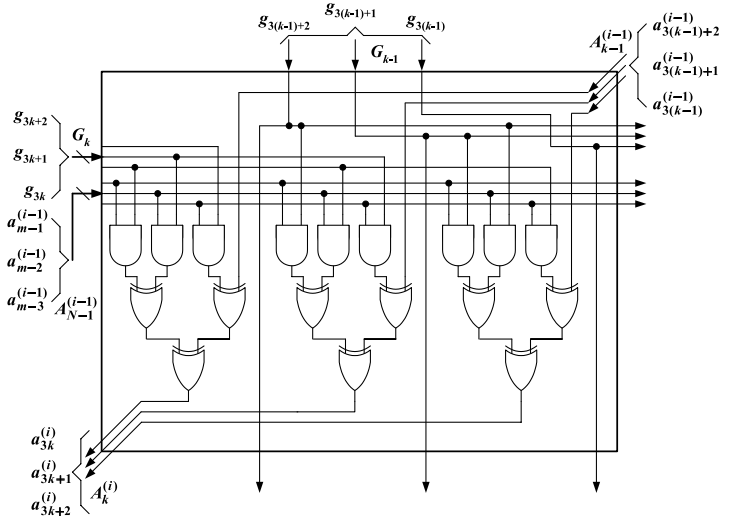


Fig. 4. Circuit of (i, k)-th basic cell in Fig. 3

$A^{(i-1)}x^D \mod G$ and $P = P^{(N)} \mod G$ are the same operation, the structure of Type-2 cell in Fig. 5 is identical with the basic cell in Fig. 3. The coefficients of the result $P(x)$ emerge from the bottom row of the array after $(N+1)$ iterations.

4 A New Digit-Serial Systolic Array for Multiplication in $GF(2^m)$

As described in Fig. 3 and Fig. 5, all the data flow is unidirectional in the horizontal direction. Therefore, we can make a projection the two DGs along the east direction following the projection procedure in [15]. Fig. 7 represents

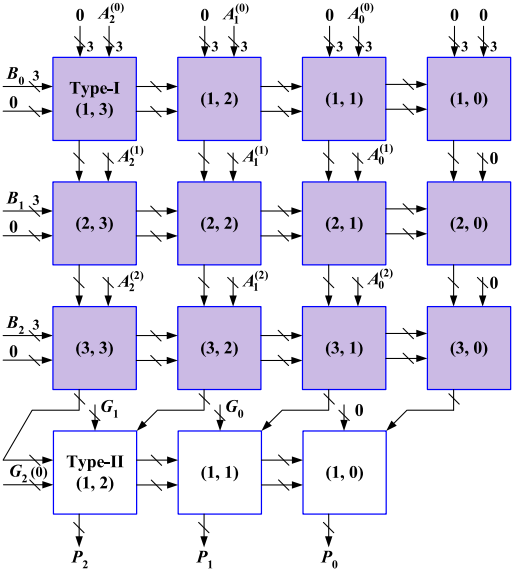


Fig. 5. Digit-level DG for $P^{(i)} = B_{i-1}A^{(i-1)} + P^{(i-1)}$ and $P = P^{(N)} \mod G$ operations in $GF(2^9)$ with $D = 3$

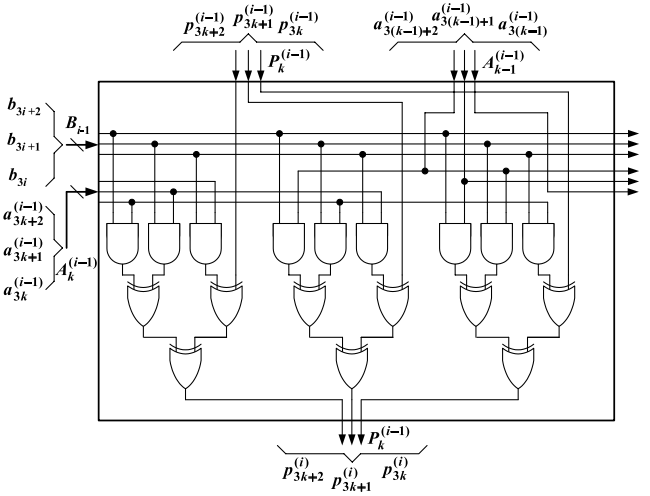


Fig. 6. Circuit of (i, k)-th basic cell in Fig. 5

one dimensional signal flow graph (SFG) array for computing $A^{(i-1)}x^D \mod G$ operation in $GF(2^9)$ with $D = 3$, where '•' denotes 1 cycle delay element. As shown in Fig. 7, it consists of $(N-1)$ units of identical processing element (PE).

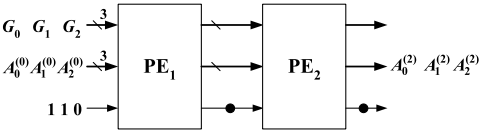


Fig. 7. One-dimensional SFG array corresponding to the DG in Fig. 3

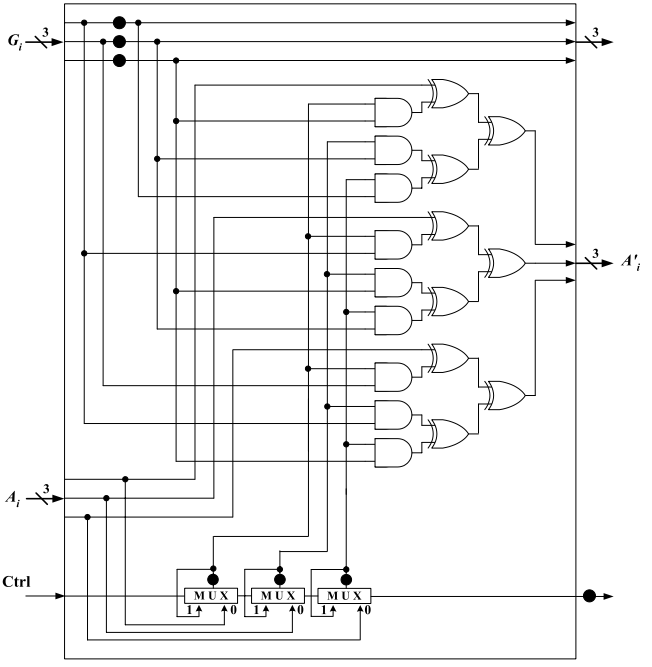


Fig. 8. Structure of each PE in Fig. 7

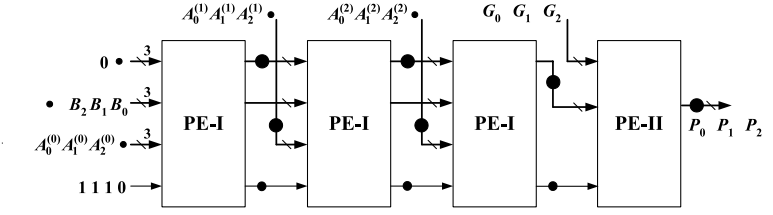


Fig. 9. One-dimensional SFG array corresponding to the DG in Fig. 5

The circuit of each PE is depicted in Fig. 8 and is controlled by a control sequence of $011\cdots1$ with length N. The digits A_i and G_i enter this array in serial form with the most significant digit first. As shown in Fig. 3, since the coefficients of $A_{N-1}^{(i-1)}$ must be broadcasted to all basic cells in the i-th row of the DG in Fig. 5, we add extra D multiplexers and D one-bit latches into each PE of

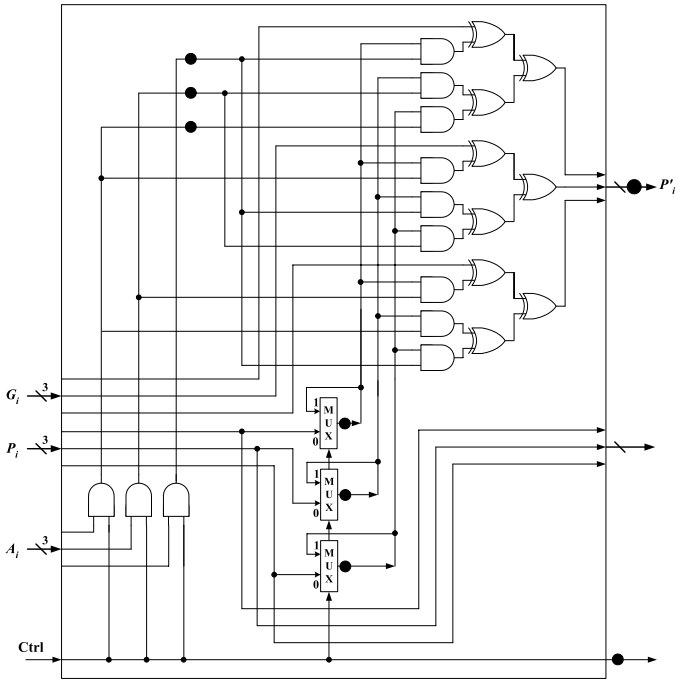


Fig. 10. Structure of each PE-I in Fig. 9

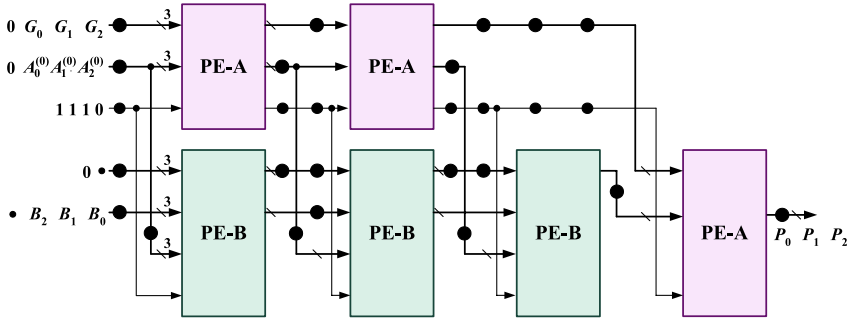


Fig. 11. A New digit-serial systolic array for multiplication in $GF(2^m)$ with $m = 9$ and $D = 3$

the SFG array in Fig. 7. When the control signal is in logic 0, the D temporary results are latched.

By applying the similar procedures, we can obtain one dimensional SFG array corresponding to Fig. 5. As described in Fig. 9, it consists of N units of identical PE-I and one PE-II. The structure of PE-I is shown in Fig. 10 and PE-II is the same circuit with the PE in Fig. 7. As described in Fig. 9, it is controlled by a control sequence of $011\cdots1$ with length $(N+1)$. We add extra D two-input

AND gates into each PE-I of the SFG in Fig. 9. This is because D constant 0 should be fed into the leftmost cell in the DG of Fig. 5.

After combining the SFG of Fig. 7 and Fig. 9, and by applying the cut-set systolisation techniques [15], we obtain a new digit-serial systolic array for multiplication in $GF(2^m)$ depicted in Fig. 11. If input data come in continuously, this array produces multiplication results at a rate of one every N clock cycles after an initial delay of $\lceil 3N \rceil + 2$ cycles. The multiplication results emerge from the right-hand side of the array in digit-serial form with the most significant digit first.

5 Performance Analysis

To verify the functionality of the proposed array in Fig 11, it was developed in VHDL and was synthesized using the Synopsis' FPGA-Express (version 2000,11-FE3.5), in which Altera's EP2A70F1508C-7 was used as the target device. After synthesizing the circuits successfully, we extract net-list files from the FPGA-Express and simulated with the net-list files using Mento graphics' design view (VHDL-ChipSim). After verifying the functionality of the proposed array in Fig 11, we compared our architecture with some related systolic arrays having the same I/O format. Table 2 summarizes the performance comparison results, which assumed that 3-input XOR gate and 4-input XOR gate are constructed using two and three 2-input XOR gates respectively. As described in Table 2, since the inner structure of the proposed array is tree-type, critical path increases logarithmically proportional to D. Therefore, the computation delay of the proposed architecture is significantly less than previously proposed digit-serial systolic multipliers whose critical path increases proportional to D.

Table 1. Comparison with previously proposed digit-serial systolic multiplier for $GF(2^m)$

	Guo et al. [12]	Kim et al. [13]	Fig. 11
Throughput	$1/N$	$1/N$	$1/N$
Latency	$3N$	$3N$	$3N+2$
Critical Path	$T_{\text{AND2}} + 3T_{\text{XOR2}} +$ $(D-1)(T_{\text{AND2}} +$ $2T_{\text{XOR2}} + T_{MUX2})$	$T_{\text{AND2}} + T_{\text{XOR2}} +$ $(D-1)(T_{\text{AND2}} +$ $T_{\text{XOR2}} + T_{MUX2})$	$T_{\text{AND2}} +$ $\lceil \log_2(D+1) \rceil T_{\text{XOR2}}$
Circuit Requirement	$\text{AND}_2 : N(2D^2+D)$ $\text{XOR}_2 : 2ND^2$ Latch : $10ND$ $\text{MUX}_2 : 2ND$	$\text{AND}_2 : N(2D^2+D)$ $\text{XOR}_2 : 2ND^2$ Latch : $10ND+N$ $\text{MUX}_2 : 2ND$	$\text{AND}_2 : N(2D^2+D)$ $\text{XOR}_2 : 2ND^2$ Latch : $10ND+2D$ $\text{MUX}_2 : 2ND$
Control Signal of	1	1	1

$N = \lceil m/D \rceil$

AND_2: 2-input AND gate

XOR_2: 2-input XOR gate

MUX_2: 2-to-1 multiplexer

T_{AND2}: propagation delay through one AND_2 gate

T_{XOR2}: propagation delay through one XOR_2 gate

T_{MUX2}: propagation delay through one MUX_2 gate

6 Conclusions

In this paper, we have proposed a new digit-serial systolic multiplier for $GF(2^m)$. From the LSD-first multiplication algorithm, we first obtained new DGs and following the projection procedure, we derived one dimensional SFG arrays and PEs, then by applying the cut-set systolisation technique, we finally constructed the digit-serial systolic multipler for $GF(2^m)$. The proposed multiplier was modeled in VHDL and simulated to verify its functionality. After verifying the proposed multiplier's functionality, we compared the performance of the multiplier with previous proposed architectures.

Two major characteristics of the proposed architecture are: 1) it has significantly less computation delay than previous architectures, and 2) if the proposed architecture is applied to ECC, which require large field size, we can select various digit size. Thus, by choosing the digit size appropriately, we can meet the throughput requirement with minimum hardware complexity. Furthermore, since the multiplier has the features of regularity, modularity, and unidirectional data flow, it is well suited to VLSI implementations.

Acknowledgement. This work was supported by grant No. R05-2003-000-11573-0 from the Basic Research Program of the Korea Science & Engineering Foundation

References

1. R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley, 1983.
2. I. F. Blake, G. Seroussi, and N. P. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, 1999.
3. S. K. Jain, L. Song, and K. K. Parhi, "Efficient Semisystolic Architectures for Finite-Field Arithmetic," *IEEE Trans. VLSI Syst.*, vol. 6, no. 1, pp. 101–113, Mar. 1998.
4. T. Zhang and K. K. Parhi, "Systematic Design Approach of Mastrovito Multipliers over $GF(2^m)$," *Proc. of the 2000 IEEE Workshop on Signal Processing Systems (SiPS): Design and Implementation*, Lafayette, LA, pp. 507–516, Oct. 2000.
5. C. S. Yeh, I. S. Reed, and T. K. Trung, "Systolic Multipliers for Finite Fields $GF(2^m)$," *IEEE Trans. Comput.*, vol. C-33, no. 4, pp. 357–360, Mar. 1984.
6. C. L. Wang and J. L. Lin, "Systolic Array Implementation of Multipliers for Finite Field $GF(2^m)$," *IEEE Trans. Circuits and Syst.*, vol. 38, no. 7, pp. 796–800, July 1991.
7. G. Orlando and C. Paar, "A Super-Serial Galois Fields Multiplier for FPGAs and its Application to Public-Key Algorithms," *Proc. of the 7th Annual IEEE Symposium on Field Programmable Computing Machines*, FCCM'99, Napa Valley, California, pp. 232–239, April. 1999.
8. M. A. Hasan and V. K. Bhargava, "Bit-Serial Systolic Divider and Multiplier for Finite Fields $GF(2^m)$," *IEEE Trans. Comput.*, vol. 41, no. 8, pp. 972–980, Aug. 1992.

9. W. C. Tsai and S. J. Wang, "Two Systolic Architectures for Multiplication in $GF(2^m)$," *IEE Proc. Comput. Digit. Tech.*, vol. 147, no. 6, pp. 375–382, Nov. 2000.
10. C. Paar, P. Fleischmann, and P. Soria-Rodriguez, "Fast Arithmetic for Public-Key Algorithms in Galois Fields with Composite Exponents", *IEEE Tans. Comput.*, vol. 48, no. 10, pp. 1025–1034, Oct. 1999.
11. L. Song and K. K. Parhi, "Low Energy Digit-Serial/Parallel Finite Field Multipliers," *J. VLSI Signal Processing*, vol. 19, no. 2, pp. 149–166, June 1998.
12. J. H. Guo and C. L. Wang, "Digit-Serial Systolic Multiplier for Finite Field $GF(2^m)$," *IEE Proc. Comput. Digit. Tech.*, vol. 145, no. 2, pp. 143–148, Mar. 1998.
13. C.H. Kim, S.D. Han and C.P. Hong, "An Efficient Digit-Serial Systolic Multiplier for Finite Fields $GF(2^m)$", *Proc. on 14th Annual IEEE International Conference of ASIC/SOC*, pp. 361–365, 2001.
14. M.C. Mekhallalati, A.S. Ashur, and M.K. Ibrahim, "Novel Radix Finite Field Multiplier for $GF(2^m)$", *J. VLSI Signal Processing*, vol. 15, no. 3, pp. . 233–245, Mar. 1998.
15. S. Y. Kung, *VLSI Array Processors*, Englewood Cliffs, NJ: Prentice Hall, 1988.
16. NIST, Recommended elliptic curves for federal government use, May 1999. http://csrc.nist.gov

Asymptotic Error Estimate of Iterative Newton-Type Methods and Its Practical Application

Gennady Yu. Kulikov[1] and Arkadi I. Merkulov[2]

[1] School of Computational and Applied Mathematics, University of the Witwatersrand,
Private Bag 3, Wits 2050, Johannesburg, South Africa
gkulikov@cam.wits.ac.za
[2] Ulyanovsk State University, L. Tolstoy Str. 42, 432970 Ulyanovsk, Russia
merkul@vda.ru

Abstract. In the paper we present a new result for evaluating the convergence error of iterative Newton-type methods with respect to the number of iteration steps. We prove an explicit asymptotically correct estimate that provide a fruitful basis to treat many practical situations. As an example of such application, we solve three important problems arising in numerical integration of ordinary differential equations and semi-explicit index 1 differential-algebraic systems.

1 Introduction

Many modern problems in applied mathematics often require to use this or that implicit computational scheme as far as such schemes are more stable than explicit ones [2]–[8]. On the other hand, any application of implicit schemes to nonlinear problems leads to systems of nonlinear equations of the form

$$x = Gx, \tag{1}$$

where G is a mapping from $\mathbf{R}^m$ to $\mathbf{R}^m$, whose exact solution cannot be found in general case. Thus, we have to apply iterative processes for solving system (1). We suppose meanwhile that the arising additional (iterative) error is sufficiently small comparing with the error of the underlying algorithm and it does not influence dramatically the accuracy of the numerical solution of the original problem. However, this assumption is true only if the number of iteration steps is great enough. A priori definition of that number is the burning task of nowadays research.

Often such iterative processes are chosen to be the simple (fixed-point) iteration, the modified or full Newton methods as well as other Newton-type iterative schemes [18]. In general, all these algorithms can be presented as follows:

$$x^N = x^{N-1} - A(x^{N-1})^{-1} F x^{N-1}, \quad N = 1, 2, \ldots, \tag{2}$$

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 667–675, 2004.
© Springer-Verlag Berlin Heidelberg 2004

where $A(x)$ is a nonsingular square matrix of dimension m, $F \stackrel{\text{def}}{=} I_m - G$, and I_m is the identity matrix of the same dimension. For example, $A \equiv I_m$ in the case of simple iteration, whenever the matrix A and the Jacobian of problem (1) ∂F coincide for the full Newton iteration and for the modified one. Interim variants are also possible, when the matrix A is more complex than the identity matrix, but is simpler than the full Jacobi matrix. It enables, on the one hand, to provide the quite high convergence of iterations (faster than in the case of fixed-point iteration steps), and, on the other hand, to simplify significantly a practical implementation and reduce execution time for such iterative schemes with respect to the full Newton method.

The Kantorovich theorem [18] gives a good estimate of the error of the full Newton iteration and enables to express explicitly this error through the parameters of the method. Therefore, in practice, it is enough to evaluate the method's parameters in order to achieve the full error value of the approximate solution. In the case of simple iteration or of modified Newton one the situation is a bit more complicated as far as their errors are approximated by some recurrently defined converging sequences [18]. However, the asymptotic error estimates of these methods obtained in [9] created the necessary theoretical basis for treating a number of problems in the area of numerical integration of ordinary differential equations and index 1 semi-explicit differential-algebraic systems (see, for example, [10], [11], [13]–[16]). Moreover, the theorem on an asymptotic estimation of the Newton-type method's error proven in [9] led to the concept of simplified Newton iterations for solving differential-algebraic systems (see [12]). Unfortunately, the result mentioned above was derived only with the accuracy of the second order terms and it does not allow the real presentation of the simplified Newton iteration convergence speed to be got for underlying discretization methods of any order greater than 1. Therefore the main goal of our paper is to specify the theorem on asymptotic error estimate of Newton-type methods and to show its application to numerical integration of both ordinary differential equations and index 1 semi-explicit differential-algebraic systems.

The paper is organized as follows: Sect. 2 is devoted to the convergence result for Newton-type iterations. Then, we show how it works in numerical solving index 1 semi-explicit differential-algebraic equations (Sect. 3), in E-methods with high derivatives for ordinary differential equations (Sect. 4) and in the numerical methods by Cash (Sect. 5). The last section summaries the results presented in this paper.

2 Asymptotic Convergence Result for Newton-Type Iterations

So, first of all we remark that the following theorem holds for all iterative schemes of the Newton type [18]:

Theorem 1 *Let the mapping $F : D \subset \mathbf{R}^m \to \mathbf{R}^m$ be Frechet differentiable on a convex set $D_0 \subset D$ and its derivative satisfies the Lipschitz condition*

$$\|\partial F(x') - \partial F(x'')\| \le \gamma \|x' - x''\| \quad \forall\ x', x'' \in D_0.$$

Suppose that $A : D_0 \subset \mathbf{R}^m \to L(\mathbf{R}^m)$ where $L(\mathbf{R}^m)$ is a space of linear operators in $\mathbf{R}^m$, and let $x^0 \in D_0$ be such that for $\delta_0, \delta_1 \ge 0$ we have

$$\|A(x) - A(x^0)\| \le \mu \|x - x^0\| \quad \forall\ x \in D_0,$$

$$\|\partial F(x) - A(x)\| \le \delta_0 + \delta_1 \|x - x^0\| \quad \forall\ x \in D_0.$$

Suppose also that matrix $A(x^0)$ is nonsingular and

$$\|A(x^0)^{-1}\| \le \beta, \quad \|A(x^0)^{-1} F x^0\| \le \eta$$

where $\beta\delta_0 < 1$, $\alpha = \sigma\beta\gamma\eta/(1-\beta\delta_0)^2 \le 1/2$, $\sigma = \max\{1, (\mu+\delta_1)/\gamma\}$. We set

$$p^* = \frac{1-\beta\delta_0}{\sigma\beta\gamma}\left(1 - (1-2\alpha)^{1/2}\right), \quad p^{**} = \frac{1-\beta\delta_0}{\beta\gamma}\left(1 + (1-2\alpha/\sigma)^{1/2}\right)$$

and assume that the closed ball $\bar{S}(x^0, p^*) \subset D_0$ where x^0 is the center of the ball and p^* is its radius. Then the iterations (2) exist, lie in $\bar{S}(x^0, p^*)$ and converge to the unique solution x^* of system (1) in $S(x^0, p^{**}) \bigcap D_0$. Moreover, the following estimate is valid:

$$\|x^* - x^N\| \le p^* - p_N, \quad N = 1, 2, \ldots,$$

where the sequence $\{p_N\}$ is defined by means of the formulas

$$p_{N+1} = p_N + \frac{1}{1-\beta\mu p_N}\left(\frac{1}{2}\sigma\beta\gamma p_N^2 - (1-\beta\delta_0)p_N + \eta\right), \quad p_0 = 0.$$

Using the technique developed in [9] (see also [15]), it is easy to prove the next theorem on asymptotic error estimate of Newton-type methods as $\alpha \to 0$.

Theorem 2 *Let the conditions of Theorem 1 hold. Then, for any sufficiently small α, the error of iteration (2) satisfies*

$$\|x^* - x^N\| \le \sum_{i=1}^{N} R_i(N)(2\alpha)^i + R_{N+1}(N)(2\alpha)^{N+1}, \quad N = 1, 2, \ldots, \tag{3}$$

where the coefficients $R_i(N)$ when $i \le N$ are defined by the recursion relation

$$R_i(N) = \beta\delta_0 R_i(N-1) + (1-\beta\delta_0) \sum_{l=1}^{i-1} \frac{\frac{1}{2}\prod_{j=1}^{l-1}(j - \frac{1}{2})}{l!} R_{i-l}(N-1), \tag{4}$$

and

$$R_{i+1}(i) = \frac{C_i}{\sigma\gamma}, \quad i = 0, 1, \ldots, N, \tag{5}$$

C_i are some positive constants which do not depend on either σ or γ.

Unfortunately, the complex form of coefficients $R_i(N)$ makes it difficult to apply Theorem 2 in practice, effectively. However, from (3)–(5) it follows

Theorem 3 *Let the conditions of Theorem 1 hold. Then, for any sufficiently small α, the error of iteration* (2) *satisfies*

$$\|x^* - x^N\| \le \frac{C}{\sigma\gamma} \sum_{i=1}^{N+1} (\beta\delta_0)^{N-i+1} (2\alpha)^i, \quad N = 1, 2, \ldots, \tag{6}$$

where C is a constant which does not depend on either σ or γ.

Note that Theorem 3 well corresponds to the previously obtained result (see Theorem 8 in [9]).

3 Simplified Newton Iteration for Index 1 Semi-explicit Differential-Algebraic Systems

Let us now discuss a practical application of Theorem 3. We start with the simplified Newton iteration suggested for solving a differential-algebraic system of the form[1]

$$x'(t) = g\big(x(t), y(t)\big), \tag{7a}$$

$$y(t) = f\big(x(t), y(t)\big), \tag{7b}$$

$$x(0) = x^0, \ y(0) = y^0 \tag{7c}$$

where $t \in [0, T]$, $x(t) \in \mathbf{R}^m$, $y(t) \in \mathbf{R}^n$, $g : D \subset \mathbf{R}^{m+n} \to \mathbf{R}^m$, $f : D \subset \mathbf{R}^{m+n} \to \mathbf{R}^n$, and the initial conditions (7c) are consistent; i. e., $y^0 = f(x^0, y^0)$. Here, it is agreed that the right-hand part of problem (7) is sufficiently differentiable, and the matrix $I_n - \partial_y f(x, y)$ is nonsingular for any $z \stackrel{\text{def}}{=} (x^T, y^T)^T \in D$, where $\partial_y f(x, y)$ denotes a partial derivative of the mapping f with respect to y.

Having used the implicit Euler method to discretize system (7) we come to

$$x_{k+1} = x_k + \tau g(x_{k+1}, y_{k+1}), \tag{8a}$$

$$y_{k+1} = f(x_{k+1}, y_{k+1}), \quad k = 0, 1, ..., K-1, \tag{8b}$$

$$x_0 = x^0, \quad y_0 = y^0 \tag{8c}$$

where τ is a step size of the numerical integration which can be fixed or variable. Then the simplified Newton iteration is formulated for (8) in the form

$$z_{k+1}^i = z_{k+1}^{i-1} - P(z_{k+1}^{i-1})^{-1} \bar{F}_k^\tau z_{k+1}^{i-1}, \quad i = 1, 2, \ldots, N, \tag{9a}$$

$$z_{k+1}^0 = \bar{z}_k = z_k^N, \quad k = 0, 1, \ldots, K-1, \tag{9b}$$

[1] Without loss of generality, we consider here only autonomous initial value problems in the class of both ordinary differential equations and semi-explicit differential-algebraic systems of index 1.

$$\bar{z}_0 = z(0) = z^0, \tag{9c}$$

where

$$P(z) \stackrel{\text{def}}{=} \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ -\frac{\partial f_1}{\partial x_1} & \cdots & -\frac{\partial f_1}{\partial x_m} & 1-\frac{\partial f_1}{\partial y_1} & \cdots & -\frac{\partial f_1}{\partial y_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -\frac{\partial f_n}{\partial x_1} & \cdots & -\frac{\partial f_n}{\partial x_m} & -\frac{\partial f_n}{\partial y_1} & \cdots & 1-\frac{\partial f_n}{\partial y_n} \end{pmatrix},$$

$\bar{F}_k^\tau z_{k+1}^{i-1} \stackrel{\text{def}}{=} I_{m+n} - \bar{G}_k^\tau z_{k+1}^{i-1}$, $\bar{G}_k^\tau z_{k+1}^{i-1}$ denotes the right-hand part of nonlinear equations (8a,b) evaluated at the point z_{k+1}^{i-1}, $\bar{z}_k = \bar{z}_k(N)$ is a value of the approximated solution of this problem at the point t_k, obtained after N iteration steps of the iterative scheme (9a) (see [12] for more detail).

Due to the special structure of matrix $P(z)$, it is evident that in the case of $m = n$ when the dimension of problem (7) is quite high the simplified Newton iteration costs about eight times cheaper from the computational point of view than the standard Newton method. Moreover, the numerical testing in [12] has exhibited that the convergence order of method (9) is not less than one shown for the modified Newton iteration. However, the result failed earlier to be proven theoretically. If we now use Theorem 3 of this paper into the proof of Theorem 1 from [12] it is easy to conclude that the combined method (9) possesses the first order convergence even when only one iteration step has been performed per each grid point. In addition, if we replace the implicit Euler method with any other one-step (or stable multistep) method of order s and apply the simplified Newton iteration to treat the resulting nonlinear system then this combined algorithm will be convergent of order $\min\{N, s\}$; i. e., the convergence orders of the simplified Newton iteration and of the modified one coincide (see [10], [11] or [15]). Thus, except the estimate of the accuracy, the new result makes it possible to define a priori a sufficient quantity of iterations per each grid point to preserve the order s of the underlying discretization method:

$$N \geq s.$$

The latter is important for an implementation of the methods to solve index 1 differential-algebraic systems (7) in practice.

4 E-Methods with High Derivatives

Another application of Theorem 3 is connected with the solution of ordinary differential equations of the form

$$x'(t) = g\big(x(t)\big), \tag{10a}$$

$$x(0) = x^0, \tag{10b}$$

where $t \in [0,T]$, $x(t) \in \mathbf{R}^m$, $g : D \subset \mathbf{R}^m \to \mathbf{R}^m$. If we extend the technique presented in [1] for the numerical integration of the initial value problem (10) to the collocation with multiple nodes (see, for example, [7]) then we result with the family of one-step two-stage methods with high derivatives:

$$x_{k+1/2} = x_k + \tau \sum_{r=0}^{p} \tau^r \left(a_1^{(r)} g_k^{(r)} + a_3^{(r)} g_{k+1}^{(r)} \right) + \tau a_2^{(0)} g_{k+1/2}, \tag{11a}$$

$$x_{k+1} = x_k + \tau \sum_{r=0}^{p} \tau^r \left(b_1^{(r)} g_k^{(r)} + b_3^{(r)} g_{k+1}^{(r)} \right) + \tau b_2^{(0)} g_{k+1/2}, \tag{11b}$$

$$x_0 = x^0, \tag{11c}$$

where $g_k^{(r)} \stackrel{\text{def}}{=} g^{(r)}(x_k)$ denotes the r-th full derivative[2] of the mapping g with respect to the independent variable t, evaluated at the point x_k, and where the coefficients of method (11) are:

$$a_1^{(r)} = \frac{p+1}{r!2^{p+r+2}} \sum_{i=0}^{p-r} \sum_{l=0}^{i+r} \sum_{j=0}^{p+1} \frac{(-1)^l (i+r)!}{l!(i+r-l)!j!(p+1-j)!(l+j+2)}$$

$$\times \sum_{q=0}^{i} \frac{(p+q)!}{q!2^q}, \quad r = 0, 1, \ldots, p,$$

$$a_2^{(0)} = \frac{(p+1)!}{2} \sum_{l=0}^{p+1} \frac{(-1)^l}{l!(p+1-l)!(2l+1)},$$

$$a_3^{(r)} = \frac{(-1)^{r+1}(p+1)}{r!2^{p+r+2}} \sum_{i=0}^{p-r} \sum_{l=0}^{i+r} \sum_{j=0}^{p+1} \frac{(-1)^j (i+r)!}{l!(i+r-l)!j!(p+1-j)!(l+j+2)}$$

$$\times \sum_{q=0}^{i} \frac{(p+q)!}{q!2^q}, \quad r = 0, 1, \ldots, p,$$

$$b_1^{(r)} = a_1^{(r)} + (-1)^r a_3^{(r)}, \quad b_2^{(0)} = 2a_2^{(0)}, \quad b_3^{(r)} = (-1)^r a_1^{(r)} + a_3^{(r)}.$$

Method (11) has stage order $2p+3$, classical order $2p+4$ and is A-stable for any integer $p \geq 0$ (all particulars will appear in [17]). Thus, it can be applied for solving many practical problems which often happen to be stiff. However, method (11) is implicit, but simple iterations are not efficient to treat stiff systems of differential equations (10) (see, for example, [2] or [8]). Therefore the usual choice is iteration (2) with the matrix A that equals to the Jacobian of the discretized system. On the other hand, it is known that the calculation of Jacobi matrix for method (11) is quite expensive because of the high derivatives. So we show how to simplify the Jacobian and retain the high order convergence of iteration (2) when the matrix A differs from the precise Jacobi matrix of discrete system (11).

[2] Here and further the zero derivative implies the original function.

Thus, we further introduce the vector

$$X_{k+1} \stackrel{\text{def}}{=} \left((x_{k+1/2})^T, (x_{k+1})^T\right)^T \in \mathbf{R}^{2m}$$

and assume that the $(2m \times 2m)$-matrix

$$A(X_{k+1}^N) \stackrel{\text{def}}{=} \begin{pmatrix} 1 - \tau a_2^{(0)} \partial g(x_{k+1/2}^N) & -\tau a_3^{(0)} \partial g(x_{k+1}^N) \\ -\tau b_2^{(0)} \partial g(x_{k+1/2}^N) & 1 - \tau b_3^{(0)} \partial g(x_{k+1}^N) \end{pmatrix}; \tag{12}$$

i. e., we have excluded all the derivatives of mapping g while evaluating the Jacobi matrix. Our aim now is to find out the convergence order of iteration (2) with matrix (12) with respect to the step size τ.

The described task is easily solved by means of Theorem 3. From (6) it follows immediately that the iterative scheme mentioned above converges with an accuracy of $O(\tau^{2N+1})$ because due to the specific form of discrete system (11) the equalities

$$\alpha = O(\tau^2),\ \beta = O(1),\ \gamma = O(\tau),\ \delta_0 = O(\tau^2),\ \sigma = O(1) \tag{13}$$

evidently hold for any sufficiently small τ. Then, taking into account the error accumulated in the course of numerical integration of differential equations (10), we conclude that the combined method (11) with the iteration (2) and matrix (12) is convergent of order $\min\{2N, 2p+4\}$. The latter implies that in order to insure the maximum order convergence it is sufficient to restrict the minimum number of iteration steps per each grid point as follows:

$$N \geq p + 2.$$

Note that the same result is valid for the modified Newton iteration (see [13] or Theorem 2.2.5 in [15]).

5 Cash's Methods

Finally, let us consider the well-known EBDFs (Extended Backward Differentiation Formulas) of Cash [5] (see also [8]). They were developed to overcome the second Dahlquist barrier. In order to increase the stability of BDFs he suggested to use an additional (off-step) point; i. e., he explored multistep formulas of the following type:

$$\sum_{i=0}^{l} a_i x_{k+1-i} = \tau b_0 g_{k+1} + \tau b g_{k+2}, \quad k = l-1, l, \ldots, K-2, \tag{14}$$

where the coefficients a_i, $i = 0, 1, \ldots, l$, b_0 and b are chosen so that method (14) is of order $l+1$ ($a_0 = 1$). The implementation of EBDF (14) is split into three stages:

1. Suppose that the solution values $x_k, x_{k-1}, \ldots, x_{k-l+1}$ are already known. Calculate $\tilde{x}_{k+1}$ by means of the BDF of order l

$$\sum_{i=0}^{l} \tilde{a}_i x_{k+1-i} = \tau \tilde{b}_0 g_{k+1}, \quad \tilde{a}_0 = 1. \tag{15}$$

2. Find $\tilde{x}_{k+2}$ as a numerical solution obtained by method (15) advanced by one step

$$\sum_{i=0}^{l} \tilde{a}_i x_{k+2-i} = \tau \tilde{b}_0 g_{k+2}. \tag{16}$$

   Here $x_{k+1} = \tilde{x}_{k+1}$.
3. Let $g_{k+2} = g(\tilde{x}_{k+2})$. Omit $\tilde{x}_{k+1}$ and, having solved (14), compute the new value for x_{k+1}.

Thus, each step of the Cash's method demands solving three nonlinear systems (14)–(16). It is obvious that the Jacobi matrices of systems (15) and (16) coincide, but the Jacobian of (14) is different. This implies an additional LU-factorization that may be costly. To avoid it Cash presented in [6] the modified EBDFs which replace method (14) with the following multistep formula:

$$\sum_{i=0}^{l} a_i x_{k+1-i} = \tau b_0 g_{k+1} + \tau(b_0 - \tilde{b}_0)\tilde{g}_{k+1} + \tau b \tilde{g}_{k+2}. \tag{17}$$

Formula (17) increases the global error of the numerical solution, but it is also of order $l+1$ and the Jacobi matrix becomes the same for all the stages of Cash's method.

Actually, there is no need in this replacement as far as in practice while solving system (14) one can apply iteration (2) with the matrix A that is just equal to the Jacobian of systems (15) and (16). Then from Theorem 3 and relations (13) it is evident that this iterative scheme converges for all nonlinear problems (14)–(16). Moreover, the order of convergence is the same in all the cases. Thus, when using estimate (6), it is easy to determine a sufficient number of iteration steps per each stage to provide the $l+1$-st order for the Cash's method mentioned above.

6 Conclusion

In this paper we have developed the new effective estimate for the error of the Newton-type iterations. We have obtained the explicit convergence result with respect to the parameters of the iterative scheme and have shown how this estimate works in practice. To the end, we remark that the application area of Theorem 3 is not bounded by the examples considered above and it can be spread into other situations when both a discretization and an iteration are needed.

References

1. Aul'chenko, S. M., Latypov, A. F., Nikulichev, Yu. V.: A method for the numerical integration of systems of ordinary differential equations using Hermite interpolation polynomials. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. **38** (1998) No. 10, 1665–1670; *translation in* Comput. Math. Math. Phys. **38** (1998) No. 10, 1595–1601
2. Arushanyan, O.B., Zaletkin, S.F.: Numerical solution of ordinary differential equations using FORTRAN. (*in Russian*) Mosk. Gos. Univ., Moscow, 1990
3. Bahvalov, N.S., Zhidkov, N.P., Kobelkov G.M.: Numerical methods. (*in Russian*) Nauka, Moscow, 1987
4. Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley and Son, Chichester, 2003
5. Cash, J.R.: On the integration of stiff systems of O.D.E.s using extended backward differentiation formulae. Numer. Math. **34** (1980) 235–246
6. Cash, J.R.: The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae. Comp. & Math. with Appls. **9** (1983) 645–657
7. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I: Nonstiff problems. Springer-Verlag, Berlin, 1987
8. Hairer, E., Wanner, G.: Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996
9. Kulikov, G.Yu.: Asymptotic error estimates for the method of simple iterations and for the modified and generalized Newton methods. (*in Russian*) Mat. Zametki. **63** (1998) No. 4, 562–571; *translation in* Math. Notes. **63** (1998) No. 3–4, 494–502
10. Kulikov, G.Yu.: Numerical methods solving the semi-explicit differential-algebraic equations by implicit multistep fixed stepsize methods. Korean J. Comput. Appl. Math. **4** (1997) No. 2, 281–318
11. Kulikov, G.Yu.: Numerical solution of the Cauchy problem for a system of differential-algebraic equations with the use of implicit Runge-Kutta methods with nontrivial predictor. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. **38** (1998) No. 1, 68–84; *translation in* Comput. Math. Math. Phys. **38** (1998) No. 1, 64–80
12. Kulikov, G.Yu.: On using Newton-type iterative methods for solving systems of differential-algebraic equations of index 1. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. **41** (2001) No. 8, 1180–1189; *translation in* Comput. Math. Math. Phys. **41** (2001) No. 8, 1122–1131
13. Kulikov G.Yu.: On implicit extrapolation methods for ordinary differential equations. Russian J. Numer. Anal. Math. Modelling. **17** (2002) No. 1, 41–69
14. Kulikov, G.Yu.: On implicit extrapolation methods for systems of differential-algebraic equations. (*in Russian*) Vestnik Moskov. Univ. Ser. 1 Mat. Mekh. (2002) No. 5, 3–7
15. Kulikov, G.Yu.: Numerical methods with global error control for solving differential and differential-algebraic equations of index 1. (*in Russian*) DCs thesis. Ulyanovsk State University, Ulyanovsk, 2002
16. Kulikov, G.Yu.: One-step methods and implicit extrapolation technique for index 1 differential-algebraic systems. Russian J. Numer. Anal. Math. Modelling. (*to appear*)
17. Kulikov, G.Yu., Merkulov, A.I.: On one-step collocation methods with high derivatives for solving ordinary differential equations. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. (*to appear*); *translation in* Comput. Math. Math. Phys. (*to appear*)
18. Ortega, J.M., Rheinboldt, W.C.: Iterative solution of nonlinear equations in several variables. Academic press, New York and London, 1970

Numerical Solution of Linear High-Index DAEs

Mohammad Mahdi Hosseini

Department of Mathematics, Yazd University, Yazd, Iran

Abstract. In this paper, a modified reducing index method is proposed for semi-explicit DAEs(differential algebraic equations)with and without constraint singularities. Also, numerical implementation of this method will be presented through pseudospectral method with and without domain decomposition. In addition, aforementioned methods will be considered by some examples.

Keywords: Differential-algebraic equations, Index reduction techniques, pseudospectral method, Domain decomposition.
AMS Subject Classification - 65L10,65L05,65L60.

1 Introduction

It is well known that the index of a differential-algebraic equations(DAEs) is a measure of the degree of singularity of the system and also widely regarded as an indication of certain difficulties for numerical methods. So, DAEs can be difficult to solve when they have a higher index, i.e., an index greater than 1 [1], and a straightforward discretization generally dose not work well. In this case, an alternative treatment is the use of index reduction methods [1,2,5,6,11], whose essence is the repeated differentiation of the constraint equations until a well-posed problem (index-1 DAEs or ordinary differential equations) is obtained. But repeated index reduction by direct differentiation leads to instability for numerical integrations (i.e., drift-off-the error in the original constraint grows). Hence, stabilized index reduction methods were used to overcome the difficulty. In [4,9], a new reducing index method has been proposed which had not need to the repeated differentiation of the constraint equations. This method has been well applied for DAEs with and without constraint singularities and the $m+1-$ index DAEs has been reducded to m-index DAEs problem. In this paper, by using the proposed method mentined in [4,9,10], the index of a DAEs (in general form) will be reduced and for instance, the 3-index Hessenberg system will be transformed to the 1-index implicit DAEs. Also, for numerical solving, pseudospectral method will be used.

It is known that the eigenfunctions of certain singular Sturm-Liouville problems allow the approximation of functions in $C^{\infty}[a,b]$ where truncation error approaches zero faster than any negative power of the number of basic functions used in the approximation, as that number (order of truncation N) tends to infinity [7]. This phenomenon is usually referred to as "spectral accuracy" [8]. The accuracy of derivatives obtained by direct, term-by-term differentiation of

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 676–685, 2004.
© Springer-Verlag Berlin Heidelberg 2004

such truncated expansion naturally deteriorates [7], but for low-order derivatives and sufficiently high order truncations this deterioration is negligible, compared to the restrictions in accuracy introduced by typical difference approximations (for more details, refer to [3,7]). Throughout, we are using first kind orthogonal Chebyshev polynomials $\{T_k\}_{k=0}^{\infty}$ which are eigenfunctions of singular Sturm-Liouville problem:

$$\left(\sqrt{1-x^2}\; T^{'}(x)\right)^{'} \; + \; \frac{k^2}{\sqrt{1-x^2}}\; T_k(x) \; = 0$$

2 DAEs with and without Constraint Singularities

Consider a linear (or linearized) model problem:

$$X^{(m)} = \sum_{j=1}^{m} A_j X^{(j-1)} + By + q \tag{1a}$$

$$0 = CX + r, \tag{1b}$$

where A_j, B and C are smooth functions of t, $t_0 \le t \le t_f$, $A_j(t) \in R^{n\times n}$, $j = 1,..,m, B(t) \in R^n, C(t) \in R^n, n \ge 2$, and CB is nonsingular (the DAE has index $m+1$) except possibly at a finite number of isolated points of t. For simplicity of exposition, let us say that there is one singularity point $t_*, t_0 < t_* < t_f$. The inhomogeneities are $q(t) \in R^n$ and $r(t) \in R$. Now, if $CB(t) \neq 0, t_0 \le t \le t_f$, then we say DAEs has not constraint singularity but DAEs has constraint singularities, if $CB(t) = 0$ at a finite number of isolated points of $t, t_0 \le t \le t_f$.

In many methods which has been used for the linear model problem (1), the following accordingly are assumed [1],

H1: The matrix function $P = B\,(CB)^{-1}\,C$ is smooth or, more precisely, P is continuous and $P^{'}$ is bounded near the singular point t_* , where we define

$$P(t_*) = \lim_{t\to t_*} (B(CB)^{-1}C)(t). \tag{2}$$

H2: The inhomogeneity $r(t)$ satisfies $r \in S$, where

$$S = \{w(t) \in R^n : \textit{there exist a smooth function } z(t) \textit{ s.t.} Cz = w \;\}$$

We note that H1 and H2 are satisfied automatically if CB is nonsingular for each t. We also indicate that here only need the continuity of P.

3 Reduction Index Method

In this section, consider DAEs (1) when $CB(t_*) = 0$. From (1.a), we can write,

$$y = (CB)^{-1}C[X^{(m)} - \sum_{j=1}^{m} A_j X^{(j-1)} - q], \qquad t \in [t_0, t_*) \cup (t_*, t_f], \tag{3}$$

which could be unbounded at the singular point t_* (whereas By is bounded). It must be mentioned that, if B has full rank y can be expressed as (3) for each $t \in [t_0, t_f]$ [1]. Now accordingly to (1), consider the following problem,

$$X^{(m)} = \sum_{j=1}^{m} A_j X^{(j-1)} + By + q, \qquad t \in [t_0, t_*) \cup (t_*, t_f] \tag{4a}$$

$$0 = CX + r, \tag{4b}$$

substituting (3) into (4.a) implies,

$$[I - B(CB)^{-1}C][X^{(m)} - \sum_{j=1}^{m} A_j X^{(j-1)} - q] = 0,$$

So, problem (4) transforms to the overdetermined system:

$$[I - B(CB)^{-1}C][X^{(m)} - \sum_{j=1}^{m} A_j X^{(j-1)} - q] = 0, \qquad t \in [t_0, t_*) \cup (t_*, t_f]$$

$$CX + r = 0.$$

The above problem, in two subdomains $[t_0, t_*)$ and $(t_*, t_f]$ will distinctly be considered, as below,

$$[I - B(CB)^{-1}C][\widehat{X}^{(m)} - \textstyle\sum_{j=1}^{m} A_j \widehat{X}^{(j-1)} - q] = 0, \qquad t \in [t_0, t_*) \tag{5a}$$

$$C\widehat{X} + r = 0, \tag{5b}$$

and

$$[I - B(CB)^{-1}C][\widetilde{X}^{(m)} - \textstyle\sum_{j=1}^{m} A_j \widetilde{X}^{(j-1)} - q] = 0, \qquad t \in (t_*, t_f] \tag{6a}$$

$$C\widetilde{X} + r = 0, \tag{6b}$$

Now, consider the overdetermined DAE system (5). Because $CB(t) \neq 0$, $t \in [t_0, t_*)$, by a simple formulation similar to the procedure mentioned in [4], this system can be transformed to a full rank DAE system with n equations and n unknowns has index m. The proofs of the below theorems and corollaries are similar to them which have been denoted in [4].

Theorem 1. Consider problem (5), when it has index two (Hessenberg system) and $n = 2$. This problem is equivalent to 1-index DAE system (7),

$$E_1 \widehat{X}' + E_0 \widehat{X} = \widehat{q}, \tag{7}$$

such that,

$$E_0 = \begin{bmatrix} b_1 a_{21} - b_2 a_{11} & b_1 a_{22} - b_2 a_{12} \\ c_1 & c_2 \end{bmatrix}, \qquad E_1 = \begin{bmatrix} b_2 & -b_1 \\ 0 & 0 \end{bmatrix},$$

$$\widehat{q} = \begin{bmatrix} -b_2 q_1 - b_1 q_2 \\ -r \end{bmatrix}, \text{ and } y = (CB)^{-1} C[\widehat{X}' - A\widehat{X} - q], \qquad t \in [t_0, t_*).$$

In the Case $n > 2$, for transforming the overdetermined system (5) to a full rank system with index m, there is a need for one additional condition on the problem (5) as below.

Suppose that there exist finite points, $t_0 < t_1 < ... < t_s < t_*, s \in N$, such that for every subinterval $(t_l, t_{l+1}), l = 0, 1, ..., s,\ t_{s+1} = t_*$,

$$\exists \quad 1 \le k_l \le n, \qquad c_{k_l}(t) \neq 0. \qquad t \in (t_l, t_{l+1}) \tag{8a}$$

Here, for simplicity let say us there is one $t_1, t_0 < t_1 < t_*$, such that

$$c_{k_0}(t) \neq 0, \quad t \in [t_0, t_1) \tag{8b}$$

and

$$c_{k_1}(t) \neq 0, \quad t \in (t_1, t_*), \tag{8c}$$

where $1 \le k_0, k_1 \le n$.

Now, we divide DAEs (5) into two problems,

$$\begin{gathered} [I - B(CB)^{-1}C][\widehat{X^1}^{(m)} - \sum_{j=1}^{m} A_j \widehat{X^1}^{(j-1)} - q] = 0, \qquad t \in [t_0, t_1) \\ C\widehat{X^1} + r = 0, \end{gathered} \tag{9}$$

and

$$\begin{gathered} [I - B(CB)^{-1}C][\widehat{X^2}^{(m)} - \sum_{j=1}^{m} A_j \widehat{X^2}^{(j-1)} - q] = 0, t \in (t_1, t_*) \\ C\widehat{X^2} + r = 0. \end{gathered} \tag{10}$$

Theorem 2. Consider the $(m+1)$-index DAE system (9), when $n > 2$ and condition (8.b) is hold, then the k_0th-row of matrix $(I - B(CB)^{-1}C)$ is linearly dependent with respect to other rows.

Now if we put $M_{n\times n} = \sum_{i=1}^{n} c_i b_i [I - B(CB)^{-1}C]$, and $\widehat{M^1}_{(n-1)\times n}$ is obtained by eliminating k_0th-row of M, then the overdetermined system (9) can be transformed to DAE system with n equations and unknowns, as below,

$$\widehat{M^1}[\widehat{X^1}^{(m)} - \textstyle\sum_{j=1}^{m} A_j \widehat{X^1}^{(j-1)} - q] = 0, \qquad t \in [t_0, t_1) \tag{11a}$$

$$C\widehat{X^1} + r = 0. \tag{11b}$$

Theorem 3. In relation (11), if $F = \begin{bmatrix} \widehat{M^1} \\ C \end{bmatrix}_{n\times n}$ and k_0 is denoted as in (8.b) then,

$$|\det F(t)| = |c_{k_0}(t)|\, |\textstyle\sum_{i=1}^{n} c_i b_i(t)|^{n-1}, \qquad t \in [t_0, t_1)$$

So, since, $\det F(t) \neq 0$, for all t in $[t_0, t_1)$, the following corollaries will obtain.

Corollary 1. Rank $(\widehat{M^1})= n-1$.

Corollary 2. The DAE system (11) is full rank.

Corollary 3. The DAE system (11) has index m.

So, through implying theorem (2) to problems (9) and (10), two DAEs (12) and (13) are obtained as below,

$$\begin{bmatrix} \widehat{M^1} \\ 0 \end{bmatrix} \widehat{X^1}^{(m)} + \begin{bmatrix} -\widehat{M^1}A_m \\ 0 \end{bmatrix} \widehat{X^1}^{(m-1)} + \cdots + \begin{bmatrix} -\widehat{M^1}A_2 \\ 0 \end{bmatrix} \widehat{X^1}' + \begin{bmatrix} -\widehat{M^1}A_1 \\ C \end{bmatrix} \widehat{X^1} = \begin{bmatrix} \widehat{M^1}q \\ -r \end{bmatrix}, \quad (12)$$

$$\begin{bmatrix} \widehat{M^2} \\ 0 \end{bmatrix} \widehat{X^2}^{(m)} + \begin{bmatrix} -\widehat{M^2}A_m \\ 0 \end{bmatrix} \widehat{X^2}^{(m-1)} + \cdots + \begin{bmatrix} -\widehat{M^2}A_2 \\ 0 \end{bmatrix} \widehat{X^2}' + \begin{bmatrix} -\widehat{M^2}A_1 \\ C \end{bmatrix} \widehat{X^2} = \begin{bmatrix} \widehat{M^2}q \\ -r \end{bmatrix}, \quad (13)$$

where $\widehat{M^1}$ and$\widehat{M^2}$ are obtained by eliminating $k_0 - th$ and $k_1 - th$ rows of the $[I - B(CB)^{-1}C]$, respectively. According to corollary 3, DAEs (12) and (13) have index m.

Now, suppose that there is one $k_2, 1 \leq k_2 \leq n$, such that

$$c_{k_2}(t) \neq 0, \quad t \in (t_*, t_f], \quad (14)$$

through implying theorem (2) to problems (6) the DAEs (14) is obtained as below,

$$\begin{bmatrix} \widetilde{M} \\ 0 \end{bmatrix} \widetilde{X}^{(m)} + \begin{bmatrix} -\widetilde{M}A_m \\ 0 \end{bmatrix} \widetilde{X}^{(m-1)} + \cdots + \begin{bmatrix} -\widetilde{M}A_2 \\ 0 \end{bmatrix} \widetilde{X}' + \begin{bmatrix} -\widetilde{M}A_1 \\ C \end{bmatrix} \widetilde{X} = \begin{bmatrix} \widetilde{M}q \\ -r \end{bmatrix}, \quad (15)$$

where $\widetilde{M}$ is obtained by eliminating $k_2 - th$ row of the $[I - B(CB)^{-1}C]$. According to theorem 3, DAEs (15) has index m.

In section 4, by using the pseudospectral method with domain decomposition numerical solution of three systems (12), (13) and (15) will simultaneously be performed and by considering initial (or boundary) conditions and continuity condition, X, in t_1 and t_* (i.e., $\widehat{X^1}(t_1) = \widehat{X^2}(t_1)$ and $\widehat{X^2}(t_*) = \widetilde{X}(t_*)$), the X and consequently, y, values will be obtained in whole interval $[t_0, t_f]$.

So, the $(m+1)$-index DAEs (4), with holding (8) and (14), can be transformed to the implicit DAEs systems (12), (13) and (15) (which have index m) by the above simple proposed formulation.

4 Implementation of Numerical Method

Here, the implementation of pseudospectral method with domain decomposition is presented for DAEs systems (12), (13) and (15) when $m = 1$ and $n = 3$. This

discussion can simply be extended to general forms. Now consider the DAEs systems,

$$\sum_{j=1}^{3} f_{ij}(t)\widehat{x_j^1}' + \sum_{j=4}^{6} f_{ij}(t)\widehat{x_{j-3}^1} = f_{i7}(t), \qquad i = 1,2 \qquad t \in [t_0, t_1) \tag{16a}$$

$$\sum_{j=1}^{3} c_j(t)\widehat{x_j^1} = -r(t), \tag{16b}$$

and

$$\sum_{j=1}^{3} g_{ij}(t)\widehat{x_j^2}' + \sum_{j=4}^{6} g_{ij}(t)\widehat{x_{j-3}^2} = g_{i7}(t), \qquad i = 1,2 \qquad t \in (t_1, t_*) \tag{17a}$$

$$\sum_{j=1}^{3} c_j(t)\widehat{x_j^2} = -r(t), \tag{17b}$$

and

$$\sum_{j=1}^{3} e_{ij}(t)\widetilde{x_j}' + \sum_{j=4}^{6} e_{ij}(t)\widetilde{x_{j-3}} = e_{i7}(t), \qquad i = 1,2 \qquad t \in (t_*, t_f] \tag{18a}$$

$$\sum_{j=1}^{3} c_j(t)\widetilde{x_j} = -r(t), \tag{18b}$$

with boundary conditions,

$$\widehat{x_1}(t_0) = \alpha_1, \tag{19a}$$

$$\widehat{x_2}(t_0) = \alpha_2, \tag{19b}$$

$$\widetilde{x_3}(t_f) = \alpha_3. \tag{19c}$$

For an arbitrary natural number ν, we suppose that the approximate solution of DAEs systems (16), (17) and (18) are as below,

$$\widehat{x_j^1}(t) = \sum_{i=0}^{\nu} a_{i+(j-1)\times(\nu+1)} T_i(\widehat{s^1}), \qquad j = 1,2,3 \qquad \widehat{s^1} \in [-1,1), \tag{20}$$

where

$$t = h^1(\widehat{s^1}) = \frac{t_1 - t_0}{2}\widehat{s^1} + \frac{t_1 + t_0}{2} \tag{21}$$

and

$$\widehat{x_j^2}(t) = \sum_{i=0}^{\nu} a_{i+3\nu+3+(j-1)\times(\nu+1)} T_i(\widehat{s^2}), \qquad j = 1,2,3 \qquad \widehat{s^2} \in (-1,1), \tag{22}$$

where

$$t = h^2(\widehat{s^2}) = \frac{t_* - t_1}{2}\widehat{s^2} + \frac{t_* + t_1}{2} \tag{23}$$

and

$$\widetilde{x_j}(t) = \sum_{i=0}^{\nu} a_{i+6\nu+6+(j-1)\times(\nu+1)} T_i(\widetilde{s}), \qquad j = 1,2,3 \qquad \widetilde{s} \in (-1,1], \tag{24}$$

where

$$t = l(\widetilde{s}) = \frac{t_f - t_*}{2}\widetilde{s} + \frac{t_f + t_*}{2}, \tag{25}$$

where $\mathbf{a} = (a_0, a_1, ..., a_{9\nu+8})^t \in R^{9\nu+9}$ and $\{T_k\}_{k=0}^{\infty}$ is sequence of Chebyshev polynomials of the first kind. Here, the main purpose is to find vector $\mathbf{a}$. Now, by using (21), we rewrite system (16) as below,

$$(\frac{2}{t_1 - t_0})\sum_{j=1}^{3} f_{ij}(h^1(\widehat{s^1}))\widehat{x_j^1}' + \sum_{j=4}^{6} f_{ij}(h^1(\widehat{s^1}))\widehat{x_{j-3}^1} = f_{i7}(h^1(\widehat{s^1})), \qquad i = 1,2, \qquad \widehat{s^1} \in [-1,1) \tag{26a}$$

$$\sum_{j=1}^{3} c_j(h^1(\widehat{s^1}))\widehat{x_j^1} = -r(h^1(\widehat{s^1})). \tag{26b}$$

Substitute (20) into (26) and by considering obtained relation, (26), substitute appropriate Chebyshev-Guass-Lobato points [6] into it (for more details refer to [2,3]). Now by repeating above procedure for systems (17) and (18), a linear system with $(9\nu + 9)$ unknowns and 9ν equations is obtained. To 9 construct the remaining equations (by attending to continuity condition of X in both points t_1 and t_* and boundary conditions (19), we put,

$\widehat{x^1}_j(1) = \widehat{x^2}_j(-1), \qquad j = 1,2,3$
$\widehat{x^2}_j(1) = \widetilde{x}_j(-1), \qquad j = 1,2,3$
$\widehat{x_1}(-1) = \alpha_1,$
$\widehat{x_2}(-1) = \alpha_2,$
$\widetilde{x_3}(1) = \alpha_3$

5 Numerical Examples

Here, we use "e_x" and "e_y" to denote the maximum absolute error in vector $X = (x_1, x_2, x_3)$ and y. These values are approximately obtained through their graphs. Results show the advantages of techniques, mentioned in sections 3 and 4. Also, the presented algorithm in section 4, is performed by using Maple V with 20 digits precision and $\lambda \geq 1$ is a parameter.

Example 1. Consider for $-1 \le t \le 1$,

$$\begin{aligned} x_1' &= -x_1 + \sin(\lambda^2 t)y + q_1(t), \\ x_2' &= -x_2 + \cos(\lambda^2 t)y + q_2(t), \\ x_3' &= -x_3 + \lambda^2 t y + q_3(t), \\ \sin(\lambda^2 t)x_1 &+ \cos(\lambda^2 t)x_2 + \lambda^2 t x_3 = -r(t), \end{aligned} \tag{27}$$

with initial conditions, $x_1(-1) = x_2(-1) = \lambda e^{-3}$, and exact solutions, $x_1(t) = x_2(t) = x_3(t) = \lambda e^{3t}$, and $y(t) = \frac{\lambda^2 e^{3t}}{4-t^2}.q(t)$ and $r(t)$ are compatible with above exact solutions. This problem has index 2. Since $c_3(t) \neq 0$, for $t \neq 0$, and $c_3(0) = 0$, hence according to condition (8) we have $t_1 = 0$, and $k_0 = k_1 = 3$. Also from (11), (12) and (15) we can convert DAEs (27) to two index 1 DAEs such that,

$$\widehat{\widehat{M}} = \widetilde{\widetilde{M}} = \begin{bmatrix} \cos^2(\lambda^2 t) + \lambda^4 t^2 & -\cos(\lambda^2 t)\sin(\lambda^2 t) & -\lambda^2 t \sin(\lambda^2 t) \\ -\cos(\lambda^2 t)\sin(\lambda^2 t) & \sin^2(\lambda^2 t) + \lambda^4 t^2 & -\lambda^2 t \cos(\lambda^2 t) \end{bmatrix}$$

In table 1, we record the results of running pseudospectral method with and without index reduction, when $\lambda = 100$.

Table 1. Maximum norm error for example 1, $\lambda = 100$

ν	Without index reduction by pseudospectral method		With index reduction by pseudospectral method with domain decomposition	
	e_x	e_y	e_x	e_y
10	80	1.3	$1.2e-5$	$4.0e-7$
15	$7.4e-4$	$1.6e-3$	$1.7e-12$	$4.0e-13$
20	$1.0e-9$	$2.5e-6$	$2.8e-15$	$5.0e-16$
25	$1.7e-13$	$3.4e-9$	$1.8e-15$	$1.0e-16$

Example 2. Consider for $0 \le t \le 1$,

$$X' = AX + By + q, \tag{28a}$$

$$0 = CX + r, \tag{28b}$$

where

$$A = \begin{bmatrix} 2 & -1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 10 \\ 10 \\ 0 \end{bmatrix},$$

$$C = \left[t^2 - (7/6)t + (1/3)\ t^2 - t/2 + 1/18\ t^2 - t + 35/144 \right],$$

with exact solutions $x_1 = x_2 = x_3 = y = \frac{\exp(3t)}{t-1.2}$ and initial conditions $x_1(0) = x_2(0) = -5/6$. $q(t)$ and $r(t)$ are compatible with above exact solutions. Here $c_1(\frac{1}{2}) = c_1(\frac{2}{3}) = 0, c_2(\frac{1}{6}) = c_2(\frac{1}{3}) = 0$ and $c_3(\frac{5}{12}) = c_3(\frac{7}{12}) = 0$. This problem has index 2. According to condition (8) we have $t_1 = \frac{5}{12}, k_1 = 1$ and $k_2 = 2$. Now, by considering matrix $M = \left[I - B(CB)^{-1}C\right]$, matrices $\widehat{\widehat{M}}$ and $\widehat{\widetilde{M}}$ (mentioned in (12) and (15)) are obtained by eliminating of first and second rows of M, respectively, as below :

$$\widehat{\widehat{M}} = \begin{bmatrix} \frac{35}{3}t - 10t^2 - \frac{10}{3} & 10t^2 - \frac{35}{3}t + \frac{10}{3} & 10t - 10t^2 - \frac{175}{72} \\ 0 & 0 & 20t^2 - \frac{50}{3}t + \frac{35}{9} \end{bmatrix}, \qquad t \in [0, \frac{5}{12}]$$

and

$$\widehat{\widetilde{M}} = \begin{bmatrix} 10t^2 - 5t + \frac{5}{9} & 5t - 10t^2 - \frac{5}{9} & 10t - 10t^2 - \frac{175}{72} \\ 0 & 0 & 20t^2 - \frac{50}{3}t + \frac{35}{9} \end{bmatrix}, \qquad t \in [\frac{5}{12}, 1]$$

Here, problem (28) with and without index reduction are solved using pseudospectral with and without domain decomposition methods. The results are represented in table 2.

Table 2. Maximum norm error for example 2

ν	Without index reduction by pseudospectral method		With index reduction by pseudospectral method with domain decomposition	
	e_x	e_y	e_x	e_y
10	1.2	0.9	$3.2e-6$	$5.6e-7$
15	$8.0e-3$	$6.7e-4$	$1.7e-10$	$5.7e-12$
20	$5.0e-8$	$2.7e-9$	$5.8e-15$	$3.8e-17$
25	$1.7e-12$	$3.4e-11$	$7.0e-18$	$1.3e-19$

The advantage of using index reduction method (proposed in sections 3 and 4) is clearly demonstrated for above example.

References

1. Ascher, U. M., Lin, P.: Sequential Regularization Methods for Higher Index Differential-Algebraic Equations with Constraint Singularities: the Linear Index-2 Case, SIAM J. Anal., Vol. 33 (1996) 1921–1940
2. Ascher, U. M., Lin, P.: Sequential Regularization Methods for Nonlinear Higher-Index DAEs, SIAM J. Sci. Comput., Vol. 18 (1997) 160–181
3. Babolian, E., Hosseini, M. M. : A Modified Spectral Method for Numerical Solution of Ordinary Differential Equations with Non-Analytic Solution, Applied Mathematics and Computation, Vol. 132 (2002) 341–351

4. Babolian, E., Hosseini, M. M. : Reducing Index, and Pseudospectral Methods for Differential-Algebraic Equations, Applied Mathematics and Computation, Vol. 140 (2003) 77–90
5. Brenan, K. E., Campbell, S.L., Petzold, L. R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, Elsevier, New York (1989)
6. Campbell, S. L.: A Computational Method for General Higher Index Singular Systems of Differential Equations, IMACS, Trans. Sci. Comput., Vol. 89 (1989) 555–560
7. Canuto, C., Hussaini, M. Y., Quarteroni, A., Zang, A.: Spectral Methods in Fluid Dynamics, Springer-Verlag (1998)
8. Gottlieb, D., Orzag, S. A.: Numerical Analysis of Spectral Methods: Theory and Applications, SIAM-CBMS, Philadelphia (1979)
9. Hosseini, M. M.: Reducing Index Method for Differential-Algebraic Equations with Constraint Singularities, J. Applied Mathematics and Computation, In Press.
10. Hosseini, M. M.: Numerical Solution of Linear Differential-Algebraic Equations, J. Applied Mathematics and Computation, In Press.
11. Wang, H., Song, Y.: Regularization Methods for Solving Differential-Algebraic Equations, Applied Mathematics and Computation, Vol. 119 (2001) 283–296

Fast Fourier Transform for Option Pricing: Improved Mathematical Modeling and Design of Efficient Parallel Algorithm

Sajib Barua, Ruppa K. Thulasiram*, and Parimala Thulasiraman

Department of Computer Science, University of Manitoba
Winnipeg, MB R3T 2N2 Canada
{sajib,tulsi,thulasir}@cs.umanitoba.ca

Abstract. Fast Fourier Transform (FFT) has been used in many scientific and engineering applications. In the current study, we have tried to improve a recently proposed model of FFT for pricing financial derivatives so as to help designing an efficient parallel algorithm. We have then developed a new parallel algorithm to compute the FFT using a swapping technique that exploits data locality, and hence showed higher efficiency of this algorithm. We have tested our algorithm on 20 node SunFire 6800 high performance computing system and compared the new algorithm with the traditional Cooley-Tukey algorithm. As an example, we have also plotted the calculated option values for various strike prices with a proper selection of log strike-price spacing to ensure fine-grid integration for FFT computation as well as to maximize the number of strikes lying in the desired region of the asset price.

Keywords: Financial Derivatives; Option Pricing; Fast Fourier Transform; Mathematical Modeling; Parallel Algorithm; Data Locality.

1 Introduction

The current state-of-the-art "grand challenges" lists problems from science and engineering [1]; some of the problems facing finance industry have recently been recognized under this grand challenges [2,3,4]. The finance industry demands efficient algorithms and high-speed computing in solving problems such as option pricing, risk analysis, and portfolio management. The solution for the optimal exercise policy must typically be performed numerically, and is usually a computationally intensive problem. To price an American option, binomial tree approach [5] has been used extensively. Recently, the option pricing problem has been studied using the Fast Fourier Transform (FFT) [6,7]. By providing an one-to-one mapping from the mathematics of Fourier space to the computational domain of the FFT, [8] explored the high performance computing for this problem. Another study [9] showed that FFT yields much better performance for the derivatives under study in comparison with the binomial lattice approach [10].

* Author for Correspondence: tulsi@cs.umanitoba.ca

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 686–695, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In the current study, we develop an improved mathematical model of FFT for option pricing and a new parallel FFT algorithm. While this new algorithm has been used in the current study for finance problem it is applicable to other scientific and engineering problems.

The rest of the paper is organized as follows. In section 2, we discuss the option pricing problem formulation using FFT. I propose the methodologies for the A description of the improvement to the mathematical modeling of FFT for option pricing is presented in section 3. In section 4, we describe the new algorithm that exploits data locality. The experimental results are presented in section 5.2 and section 6 concludes.

2 Mathematical Modeling of FFT for Option Pricing

Following the Carr and Madan's work [6] on the use of FFT for option pricing, we write the call price function as

$$C_T(k) = \frac{exp(-\alpha k)}{\pi} \int_0^\infty e^{-ivk} \psi_T(v) dv. \tag{1}$$

where $\psi_T(v)$ is the Fourier transform of this call price $C_T(k)$ given by

$$\begin{aligned} \psi_T(v) &= \int_{-\infty}^{\infty} e^{ivk} c_T(k) dk\,, \\ &= \int_{-\infty}^{\infty} e^{ivk} e^{-rT} \int_k^\infty e^{\alpha k} (e^s - e^k) q_T(s) ds dk. \end{aligned} \tag{2}$$

$$\psi_T(v) = \frac{e^{-rT} \phi_T(v - (\alpha + 1)i)}{\alpha^2 + \alpha - v^2 + i(2\alpha + 1)v}. \tag{3}$$

$\psi_T(v)$ is odd in its imaginary part and even in its real part. Here k is the log strike price K ($k = \log(K)$). The call option price needs to be computed at various strike prices of the underlying assets in the option contract. q_T(s) in (3) is the risk-neutral density of the pricing model. The integration in (1) is a direct Fourier transform and lends itself to the application of FFT. If $M = e^{-\alpha k}/\pi$ and $\omega = e^{-i}$ then

$$C_T(k) = M \int_0^\infty \omega^{vk} \psi_T(v) dv. \tag{4}$$

If $v_j = \eta(j-1)$ and trapezoid rule is applied for the integral on the right of (4), $C_T(k)$ can be written as

$$C_T(k) \approx M \sum_{j=1}^{N} \psi_T(v_j) \omega^{v_j k} \eta,\ k = 1, \ldots, N. \tag{5}$$

where the effective upper limit of integration is $N\eta$ and v_j corresponds to various prices with η spacing.

3 Improvement to the Mathematical Modeling

Most recent research on option valuation has successfully applied Fourier analysis to calculate option prices. As shown above in (1), to obtain the analytically solvable Fourier transform, the call price function needs to be multiplied with an exponential factor, $e^{\alpha k}$ ($c_T(k) = e^{\alpha k} C_T(k)$). The calculation of $\psi_T(v)$ in (3) depends on the factor $\phi_T(u)$, where $u = v - (\alpha + 1)i$. The calculation of the intermediate function $\phi_T(u)$ requires specification of the risk neutral density function, $q_T(s)$. The limits on the integral have to be selected in such a way as to generate real values for the FFT inputs. To generate the closed form expression of the integral, the integrands, especially the function $q_T(s)$, have to be selected appropriately. Without loss of generality we use uniform distribution for $q_T(s)$. This implies occurrence of a range of terminal log prices at equal probability, which could, of course, be relaxed and a normal distribution could be employed. Since the volatility is assumed constant (low) the variation in the drift is expected to cause a stiffness in the system. However, since we have assumed uniform distribution for $q_T(s)$, variation in drift is eliminated and hence the stiffness is avoided. For numerical calculation purposes, the upper limit of (4) is assumed as a constant value and the lower limit is assumed as 0. The upper limit will be dictated based on the terminal spot price. In other words, to finish the call option in-the-money the upper limit will be smaller than the terminal asset price. Therefore, the equation is:

$$\phi_T(u) = \int_0^{\lambda} e^{ivk} q_T(s) ds = \int_0^{\lambda} (\cos(vk) + i \sin(vk)) q_T(s) ds. \tag{6}$$

Without loss of generality, modifications are required as derived below. The purpose of these modifications is to generate feasible and tractable initial input condition to the FFT algorithm from these equations. Moreover, these modifications make the implementation easier.

$$\psi_T(v) = \int_{-\alpha}^{\alpha} e^{ivk} e^{-rT} \int_k^{\alpha} e^{\alpha k} (e^s - e^k) q_T(s) ds dk\,, \tag{7}$$

$$= \frac{e^{-rT} \Phi_T(v - (\alpha + 1)i)}{\alpha^2 + \alpha - v^2 + i(2\alpha + 1)v}\,,$$

$$= \frac{e^{-rT} \Phi_T(v - (\alpha + 1)i)((\alpha^2 + \alpha - v^2) - i(2\alpha + 1)v)}{((\alpha^2 + \alpha - v^2)^2 + (2\alpha + 1)^2 v^2)}. \tag{8}$$

Now,

$$\Phi_T(u) = \int_0^{\lambda} e^{ius} q_T(s) ds \tag{9}$$

where λ is terminal spot price and integration is taken only in the positive axis. To calculate $\phi_T(v - (\alpha + 1)i)$, $v - (\alpha + 1)i$ is substituted by u in (9) which gives:

$$\phi_T(v - (\alpha + 1)i) = \int_0^{\lambda} e^{(iv + \alpha + 1)s} q_T(s) ds\,, \tag{10}$$

assuming $q_T(s)$ as an uniform distribution function of the terminal log price, this can be shown as

$$= \frac{q_T(s)}{(\alpha+1)^2+v^2}\Big[[e^{(\alpha+1)\lambda}\{(\alpha+1)\cos(\lambda v)+v\sin(\lambda v)\}-(\alpha+1)] + i[e^{(\alpha+1)\lambda}\{(\alpha+1)\sin(\lambda v)-v\cos(\lambda v)\}+v]\Big]. \quad (11)$$

If we assume $e^{(\alpha+1)\lambda}\{(\alpha+1)\cos(\lambda v)+v\sin(\lambda v)\}-(\alpha+1)=\Delta$ and $e^{(\alpha+1)\lambda}\{(\alpha+1)\sin(\lambda v)-v\cos(\lambda v)\}+v=\Delta_x$ then (11) can be simplified as

$$\phi_T(v-(\alpha+1)i)=\frac{q_T(s)}{(\alpha+1)^2+v^2}(\Delta+i\Delta_x). \quad (12)$$

Substituting (12) in (8) gives

$$\psi_T(v)=\frac{e^{-rT}q_T(s)}{\{(\alpha+1)^2+v^2\}\{(\alpha^2+\alpha-v^2)^2+(2\alpha+1)^2v^2\}}\times \Big[\{(\alpha^2+\alpha-v^2)\Delta+(2\alpha+1)v\Delta_x\}+i\{(\alpha^2+\alpha-v^2)\Delta_x-(2\alpha+1)v\Delta\}\Big]. \quad (13)$$

We use this final expression for the new parallel FFT algorithm to compute the call price function. The financial input data set for our parallel FFT algorithm is the calculated data points of $\psi_T(v)$ for different values of v.

We calculate call value for different strike price values v_j where j will range from 1 to N. The lower limit of strike price is 0 and upper limit is $(N-1)\eta$ where η is the spacing in the line of integration. Smaller value of η gives fine grid integration and a smooth characteristics function of strike price and the corresponding calculated call value.

The value of k on the left of (5) represents the log of ratio of strike and terminal spot price. The implementation of FFT mathematical model returns N values of k with a spacing size of γ and these values are fed into a parallel algorithm to calculate N values of $C_T(k)$. Here, we consider cases in the range of in-the-money to at-the-money call values. The value of k will be 0 for at-the-money call - that is strike price and exercise price are equal. The value of k will be negative when we are in-the-money and positive when we are out-of-the-money. If γ is the spacing in the k then the values for k can be obtained from the following equation:

$$k_u=-p+\gamma(u-1), \text{ for } u=1,\ldots,N. \quad (14)$$

So the log of the ratio of strike and exercise price will range from $-p$ to p where $p=\frac{N\gamma}{2}$. Substitution of (14) in (5) will give

$$C_T(k_u)\approx\frac{exp(-\alpha k_u)}{\pi}\sum_{j=1}^{N}e^{-iv_j(-p+\gamma(u-1))}\psi_T(v_j)\eta, \text{ for } u=1,\ldots,N. \quad (15)$$

Replacing v_j with $(j-1)\eta$ in (15), we get

$$C_T(k_u) \approx \frac{exp(-\alpha k_u)}{\pi} \sum_{j=1}^{N} e^{-i\gamma\eta(j-1)(u-1))} e^{ipv_j} \psi_T(v_j)\eta, \text{ for } u = 1, \ldots, N. \quad (16)$$

The basic equation of FFT is

$$Y(k) = \sum_{j=1}^{N-1} e^{-i\frac{2\pi}{N}(j-1)(k-1)} x(j), \text{ for } k = 1, \ldots, N. \quad (17)$$

Comparing the above equation with the basic equation of FFT we can note that $\gamma\eta = \frac{2\pi}{N}$.

In our experimental result of 1024 (N) number of calculated call values, assuming $\eta = 0.25$ with the intuition that it will ensure fine grid integration, γ is calculated as 0.02454.

4 Parallel FFT Algorithm Exploiting Data Locality

Figure 1 illustrates the Cooley-Tukey algorithm [11] and the *butterfly* computation. Let us assume we have N ($N = 2^m$) data elements and P ($P = 2^p$) processors where $N > P$. A butterfly computation is performed on each of the data points in every iteration where there are $\frac{N}{2}$ summations and $\frac{N}{2}$ differences. The FFT is inherently a synchronous algorithm. In general, a parallel algorithm for FFT with blocked data distribution [12] where $\frac{N}{P}$ data is allocated to every processor involves communication for $\log P$ iterations and terminates after $\log N$ iterations. The input data points are bit reversed before feeding to the parallel FFT algorithm. If we assume shuffled input data at the beginning, the first $\log N - \log P$ stages require no communication. That is, the data required for the butterfly computation, resides in each local processor. Therefore, during the first $(\log N - \log P)$ iterations, a sequential FFT algorithm can be used inside each processor (called local algorithm).

At the end of the $(\log N - \log P)th$ iteration, the latest computed values for $\frac{N}{P}$ data points exist in each processor. The last $\log P$ stages require remote communications (called remote algorithm). The partners of each of the $\frac{N}{P}$ data points in processor P_i required to perform the actual butterfly computation at each iteration reside in a different processor P_j. In a blocked data distribution, therefore, $\frac{N}{P}$ amount of data is communicated by each processor for $\log P$ stages. The message size is $\frac{N}{P}$.

In Fig. 1 a), we can see that calculating Y_0 in processor 0 requires two data points, one of which reside in the local processor (=0), and the other resides in processor 2, and hence requires one communication to calculate Y_0. Similarly, calculating Y_1, Y_2, and Y_3 need 3 more communications with processor 2. Each processor requires 4 communications to calculate 4 FFT output. In total, 16 communications are required.

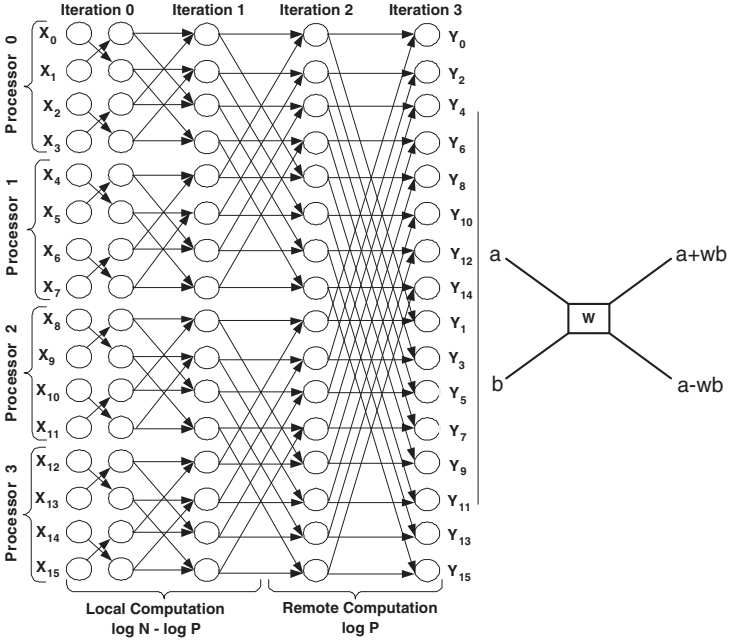


Fig. 1. a) Cooley-Tukey Algorithm, and b) Butterfly operation.

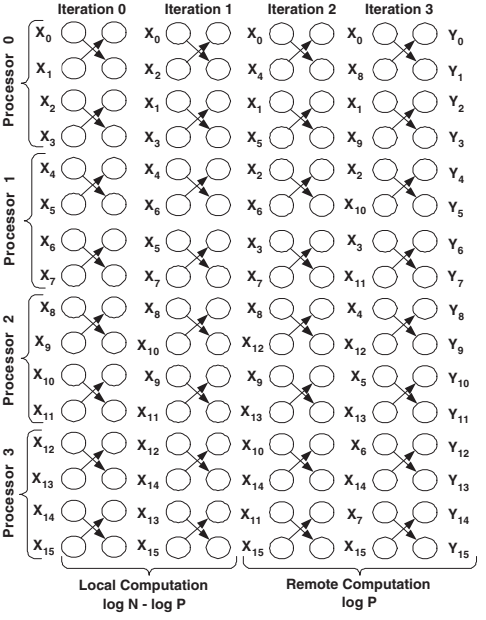


Fig. 2. Data Swapping Algorithm.

In our data swapping algorithm, depicted in Fig. 2, we apply the same blocked data distribution and the first $(\log N - \log P)$ stages require no communication. However, in the last $\log P$ stages that require communication, we swap some data at each stage and let the data reside in the processor's local memory after swapping. Therefore, the identity of some of the data points in each processor changes at every stage of the $\log P$ stages.

In Fig. 2, we can see that calculating the first two output data points in processor 0 needs two input data points with index 0 and 8 and node with index 8 does not reside in the local processor. So we need one communication to bring node 8 from processor 2. Similarly, calculating the next two output data points need one more communication. Therefore, in processor 0, we need two communications to calculate four output data points. With the same arguments, each of the processors 1, 2, and 3 needs 2 communications. In total, 8 communications are required to calculate FFT of 16 data points. So in the new parallel FFT algorithm, the number of communications is reduced by half.

We take advantage of the the fact that communication between processors is point to point and swap the data in a similar manner. However, in this case, only $\frac{N}{2P}$ amount of data (message size) is communicated by each processor at every stage. Also note that, data swapping between processors at each location allows both the upper and lower part of the butterfly computations to be performed locally by each processor. This improvement enhances good data locality and thereby providing performance increase in the new FFT algorithm compared to the Cooley-Tukey algorithm.

5 Experimental Results

In this paper, we concentrate on the performance of the new algorithm as implemented on a distributed memory environment. However, we present one computational result on the call values using FFT in the following subsection and the performance results elaborately in section 5.2.

5.1 Computational Results – An Example

The Fig. 3 depicts the calculated in-the-money call values for different strike prices and it shows that the normalized option value is decreasing with the increase of strike price. If X is the strike price and S_T is the terminal spot price of the underlying asset, the European call value is $max(S_T - X, 0)$. In this experiment, strike price can be any value between 0 and 300. Since we are considering in-the-money call, the terminal spot price is greater than the strike price. For this particular experiment, (with $\eta = 0.25$, $\gamma = 0.02454$, and $N = 1024$) the terminal spot price is 127 and to calculate in-the-money call, the strike price range from 0 to 150. With the increase of strike price from 0 towards 127, $(S_T - X)$ is supposed to decrease, which can be seen in the Fig. 3. However, we report the results of the performance experiment.

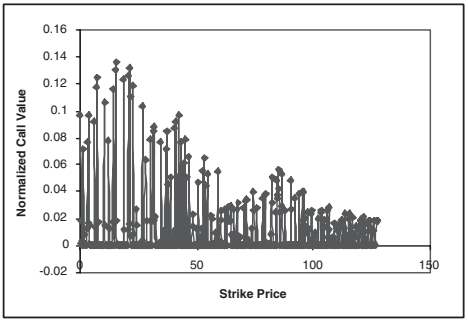


Fig. 3. Computed Call values.

5.2 Performance Results

The experiments were conducted on a 20 node SunFire 6800 high performance computing system at the University of Manitoba running MPI. The Sunfire consists of Ultra Sparc III CPUs, with 1050 MHz clock rate and 40 gigabytes of memory and runs Solaris 8 operating system. The data generated in section 3 is used for the FFT input. Figure 4 depicts a comparison of the execution time between the swap algorithm and the Cooley-Tukey algorithm. At each iteration $\frac{N}{2P} = \frac{2^{20}}{2^5} = 2^{15}$ data points are swapped on each of the 16 processors. On a 2 processor machine, there are $\log 2^{20} - \log 2 = 19$ local computations and only 1 remote communication. However, there is a significant decrease in execution time in 16 processors. This is attributed to the fact that in MPI, the packing and unpacking of $\frac{N}{2P} = 2^{18}$ data elements for each of the 2 processors requires significant amount of time.

When we compare the swap algorithm to the Cooley-Tukey algorithm in Fig. 4 on 16 processors, the swap algorithm performs 15% better than Cooley-Tukey algorithm on a data size of 2^{20}.

We calculated the efficiency of the swap algorithm for various processors on a fixed data size as presented in Fig. 5. The efficiency for 16 processors is close to 1. For 4, 8, and 16 processors the efficiency is 90% for data sizes 2^{14}, 2^{16}, 2^{19} respectively. Also for 8 and 16 processors the efficiency is 50% for 2^{12} and 2^{13} respectively. These results illustrate that as we increase the data size and the number of processors, the swap algorithm exhibits very good scalability.

Figure 6 compares the speedup results of both the swap and Cooley-Tukey algorithms. The speedup of the swap algorithm for data sizes 2^{16} and 2^{19} for large number of processors produce better results than the Cooley-Tukey algorithm.

6 Conclusions

In this paper, without loss of generality, we have improved the mathematical modeling of FFT for option pricing and we have identified appropriate values

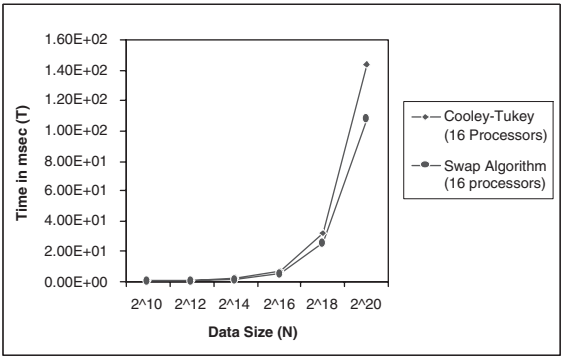


Fig. 4. Comparison of the execution times of swap and Cooley-Tukey algorithms.

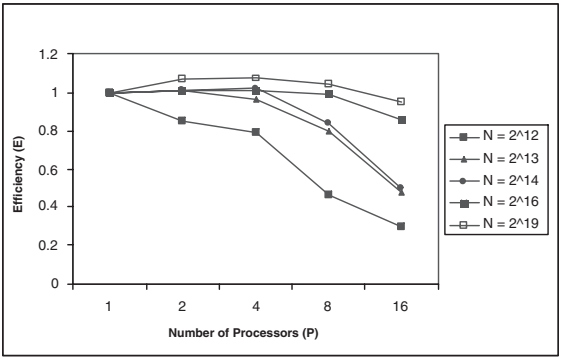


Fig. 5. Efficiency of the swap algorithm.

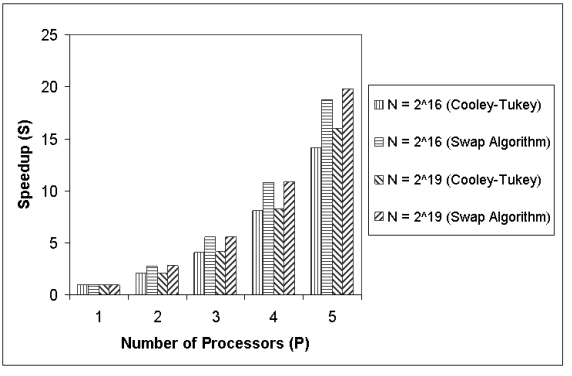


Fig. 6. Comparison of the speedup for Cooley-Tukey and swap algorithms.

for the parameters to generate the input data set for the parallel FFT computations. A basic parallel implementation of FFT on a distributed platform using MPI for message passing was carried out first. The communication latency was reduced by improving the data locality, a main challenge in developing the new parallel FFT algorithm. We have integrated the improved mathematical model to the new parallel FFT algorithm and studied the performance results. Compared to the traditional Cooley-Tukey algorithm, the current algorithm with data swapping performs better by more than 15% for large data sizes.

Acknowledgements. The authors gratefully acknowledge partial financial support from Natural Sciences and Engineering Research Council (NSERC) of Canada and the University of Manitoba Research Grant Program (URGP).

References

1. A. B. Tucker. *Computer Science and Engineering Handbook.* CRC Press, Boca Raton, Florida, 1997.
2. M. B. Haugh and A. W. Lo. Computational Challenges in Portfolio Management - Tomorrow's Hardest Problem. *Computing in Science and Engineering*, 3(3):54–59, May-June 2001.
3. E. J. Kontoghiorghes, A. Nagurnec, and B. Rustem. Parallel Computing in Economics, Finance and Decision-making. *Parallel Computing*, 26:207–209, 2000.
4. S. A. Zenios. High-Performance Computing in Finance: The Last 10 Years and the Next. *Parallel Computing*, 25:2149–2075, Dec. 1999.
5. John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option Pricing: A Simplified Approach. *Journal of Financial Economics*, 7:229–263, 1979.
6. P. Carr and D. B. Madan. Option Valuation using the Fast Fourier Transform. *The Journal of Computational Finance*, 2(4):61–73, 1999.
7. M.A.H. Dempster and S.S.G Hong. Spread Option Valuation and the Fast Fourier Transform. Technical Report WP 26/2000, Judge Institute of Management Studies, Cambridge, England, 2000.
8. R. K. Thulasiram and P. Thulasiraman. Performance Evaluation of a Multithreaded Fast Fourier Transform Algorithm for Derivative Pricing. *The Journal of Supercomputing*, 26(1):43–58, Aug. 2003.
9. R. K. Thulasiram and P. Thulasiraman. A Parallel FFT Approach for Derivative Pricing. In *Proceedings of SPIE Vol.4528: Commercial Applications for High Performance Computing; (Ed.:H.J. Siegel)*, pages 181–192, Denver, CO, Aug. 2001.
10. R. K. Thulasiram, L. Litov, H. Nojumi, C. T. Downing, and G. R. Gao. Multithreaded Algorithms for Pricing a Class of Complex Options. In *Proceedings (CD-ROM) of the International Parallel and Distributed Processing Symposium(IPDPS)*, San Francisco, CA, Apr. 2001.
11. J.W. Cooley, P.A. Lewis, and P.D. Welch. *The Fast Fourier Transform and its Application to Time Series Analysis.* Wiley, New York, 1977. In statistical Methods for Digital Computers.
12. A. Grama and A. Gupta and V. Kumar and G. Karypis. *Introduction to Parallel Computing.* Pearson Educarion Limited, Edinburgh Gate, Essex, Second edition, 2003.

Global Concurrency Control Using Message Ordering of Group Communication in Multidatabase Systems

Aekyung Moon[1] and Haengrae Cho[2]

[1] Software Robot Research Team, ETRI,
Gajung-dong, Yusong-gu, Taejon 305-350, Republic of Korea
akmoon@etri.re.kr
[2] Department of Computer Engineering, Yeungnam University
Kyungsan, Kyungbuk 712-749, Republic of Korea
hrcho@yu.ac.kr

Abstract. A multidatabase system (MDBS) is designed to provide universal access to distributed data across multiple autonomous and possibly heterogeneous local database systems. In this paper, we propose a new global concurrency control algorithm, named GCC-M, in the MDBS. GCC-M is the first algorithm that integrates the idea of message ordering of group communication to the global concurrency control. The message ordering allows to determine easily the relative serialization order of global transactions, and thus it can make the global concurrency control algorithm be simple and can also avoid distributed deadlocks. Using a distributed database simulation model, we show that GCC-M outperforms the previous global concurrency control algorithms in a wide variety of database workloads.

1 Introduction

A *multidatabase system* (MDBS) is designed to provide distributed data access across multiple autonomous and possibly heterogeneous *local database systems* (LDBSs) [1]. As providing universal access to countless sources of heterogeneous LDBSs through network, the MDBS is expected to play a key role in the future of information management.

In the MDBS, a *global concurrency control* (GCC) algorithm is required to ensure data consistency in the presence of global transactions, which may execute at several LDBSs through the MDBS interface. A major issue in designing the GCC algorithm is to handle the autonomy and heterogeneity of each LDBS [11]. Unlike traditional distributed database systems, where concurrency control is carried out through a common protocol embedded in every LDBS, the MDBS can not rely on obtaining the control information from LDBSs for GCC. Existing GCC algorithms can support local autonomy or not [1]. The former might result in poor performance due to either a low degree of concurrency or high transaction abort ratio [3,8]. The latter restricts the type of participating LDBSs or partitions the database of each LDBS [2,12].

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 696–705, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this paper, we propose a new GCC algorithm for ensuring global serializability and local autonomy in MDBS. A necessary condition of GCC algorithm is that all global transactions are serialized in the same order at all LDBSs that they execute. To achieve this, we take advantage of *message ordering property* [9,10] on the channel between LDBSs and MDBS. If all operations of a transaction are bundled in a single message and the message arrives at LDBSs in the same order using the message ordering property, each LDBS can perform subtransactions in the same order. As a result, it is possible to determine easily the relative serialization order of global transactions without any execution information of LDBSs.

This paper is organized as follows: Sect. 2 presents the related work. Sect. 3 describes the details of the proposed algorithm and its effectiveness is discussed in Sect. 4. In Sect. 5, we first present an experiment model to evaluate the performance of the proposed algorithm and then analyze the experiment results. Finally, Sect. 6 summarizes the main conclusions of this study.

2 Related Work

In this section, we first present previous GCC algorithms and then introduce the notion of message ordering.

The representative GCC algorithms supporting local autonomy are optimistic ticket method (OTM) [8] and chain-conflicting serializability [16]. Both algorithms force conflicts between global transactions. Specifically, OTM uses a *ticket* whose value is stored as a regular data item in each LDBS. Then OTM can create direct conflicts between global transactions by requiring their subtransactions to issue a take-a-ticket operation that consists of reading the value of ticket and updating it. Note that OTM should suffer from high transaction abort ratio or low concurrency due to frequent conflicts on the ticket.

A message ordering is a software that allows two entities to communicate with preserving the order of sending and receiving messages. Specifically, messages sent by a given node are delivered at all nodes in the order they were sent. Furthermore, messages from different nodes are delivered in the same total order at all sites. Using the message ordering property, it is possible to build the reliable delivery mechanism for distributed enterprize applications and Internet-based business solutions [4,14]. Recently a number of commercial message ordering products have been provided: IBM's MQSeries, Progress's SonicMQ, Fiorano's FioranoMQ, Sun's JMQ, and so on [7,14].

The message ordering property can alleviate the complexity of GCC algorithms. If all operations of a transaction are bundled in a single message, the message ordering guarantees that the messages are delivered to the participating LDBSs in the same order. This means that the message ordering can control the relative serialization order of global transactions at all LDBSs that they execute. The same execution order can also avoid distributed deadlocks [9,10]. However, these novel advantages can only be achieved if every LDBS should execute global transactions according to their delivery order.

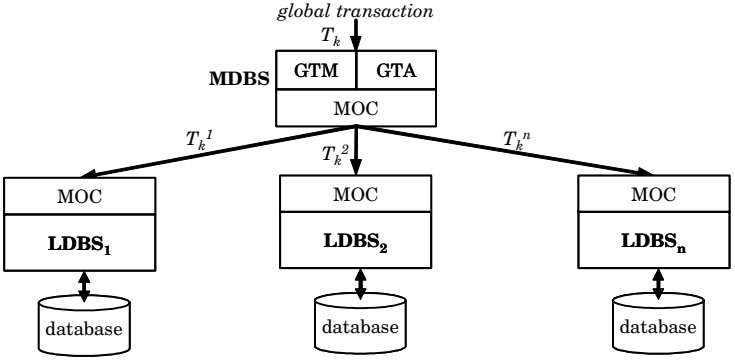


Fig. 1. Multidatabase system architecture

3 Proposed Concurrency Control Algorithm

We propose a new GCC algorithm, named *Global Concurrency Control algorithm based on Message ordering* (GCC-M). GCC-M can exploit the potential advantages of message ordering by forcing each LDBS to preserve the delivery order without violating local autonomy.

The MDBS architecture (Fig. 1) has three components: global transaction manager (GTM), global transaction agent (GTA) and message ordering component (MOC). The GTM is responsible for accepting global transactions and modelling their execution. A global transaction is decomposed into *subtransactions*, each of which accesses disjoint LDBSs in parallel. The GTM checks whether there is a direct conflict between subtransactions by maintaining two information: *transaction table* (TTBL) and *message queue* (MQ). For a i^{th} subtransaction (T_k^i) of a global transaction T_k, the TTBL registers $[RS_k^i, WS_k^i]$, where RS_k^i is the read set of T_k^i and WS_k^i is the write set of T_k^i. MQ[i] registers uncommitted subtransactions accessing LDBS$_i$. If T_{k-1}^i precedes T_k^i in MQ[i], then the serialization order of T_{k-1} precedes that of T_k. The MOC guarantees the message ordering of $T_{k-1} \rightarrow T_k$ by providing common interface or wrapper for existing LDBSs. Specifically, the sender part of MOC attaches a unique timestamp for each subtransaction and the receiver part of MOC puts the subtransaction into a local queue ordered according to its timestamp [15]. We use the term MOC$_i$ to denote the receiver part of MOC at LDBS$_i$.

The GTM forwards subtransactions of T_k to the GTA allocated for T_k. Then the GTA delivers subtransactions to LDBSs through the MOC. MOC$_i$ issues operations of subtransactions one after another into LDBS$_i$ so that all LDBSs create the same execution order. Note that a serial execution of global transactions by their delivery order may not guarantee global serializability due to the indirect conflicts incurred by local transactions. GCC-M forces conflicts between global transactions similar to OTM [8] but more efficiently. The followings describe the procedure of GCC-M in detail.

Step 1: The GTM decomposes a global transaction T_k into subtransactions so that a subtransaction T_k^i accesses an LDBS$_i$. For each T_k^i, the GTM registers $[RS_k^i, WS_k^i]$ into TTBL$[k][i]$ and inserts T_k^i into MQ$[i]$.

Step 2: Then the GTM validates the relationship of T_k^i and a subtransaction just in front of T_k^i in MQ$[i]$. Suppose T_{k-1}^i precedes T_k^i in MQ$[i]$. Then the serialization order of global transactions is $T_{k-1} \rightarrow T_k$. T_{k-1}^i and T_k^i are *direct conflict* if a data item in the write set of one subtransaction appears in the read set or write set of another subtransaction. Let the *conflict information* be a pair of operations (o_{k-1}^i, o_k^i) which incur the conflict between T_{k-1}^i and T_k^i. If T_{k-1}^i and T_k^i are not direct conflict, the GTM has to assign additional operation to force conflicts between them as follows:

1. If WS_{k-1}^i is not empty, select a data item $x \in WS_{k-1}^i$. Then append a conflict operation of reading x into the last position of T_k^i.
2. If WS_{k-1}^i is empty, select a data item $x \in RS_{k-1}^i$. Then append a conflict operation of updating x into the last position of T_k^i. In this case, the updated value of x is equal to the original one.

Step 3: Once the conflict information is defined for every subtransaction of T_k, the GTA delivers subtransactions of T_k together with their conflict information to the corresponding receiver parts of MOC.

Step 4: On receiving T_k^i and its conflict information, MOC$_i$ issues operations of T_k^i one after another into LDBS$_i$ until its conflict operation of o_k^i is met. When LDBS$_i$ responds the result of each operation, MOC$_i$ marks that the operation is executed. In case of o_k^i, MOC$_i$ first checks whether the preceding conflict operation of o_{k-1}^i was marked to have been executed. If o_{k-1}^i was executed, MOC$_i$ issues o_k^i and thus T_k^i is blocked until T_{k-1}^i commits in LDBS$_i$. Otherwise, issuing o_k^i would violate the serialization order; hence, MOC$_i$ delays o_k^i until o_{k-1}^i is executed. This may introduce a local deadlock involving MOC$_i$ and LDBS$_i$. So a timeout mechanism is required to prevent delaying o_k^i permanently.

To illustrate the procedure of GCC-M, consider two local database systems LDBS$_1$ and LDBS$_2$. Suppose that LDBS$_1$ stores a data item a, and LDBS$_2$ stores data items b and c. Suppose also that there are two global transactions G_1 and G_2, and a local transaction T_3. G_1 consists of two operations $r_1(a)$ $w_1(c)$, while G_2's operations are $w_2(a)$ $r_2(b)$. T_3 consists of $r_3(c)$ $w_3(b)$. The following shows an incorrect schedule violating global serializability.

$$\text{LDBS}_1 : r_1(a)\ w_2(a),\ G_1 \rightarrow G_2$$
$$\text{LDBS}_2 : r_3(c)\ w_1(c)\ r_2(b)\ w_3(b),\ G_2 \rightarrow T_3 \rightarrow G_1$$

GCC-M can resolve the problem as follows. The GTM first decomposes G_1 and G_2, and then registers $[RS_1^1 = \{a\}, WS_1^1 = \emptyset]$ for G_1^1 and $[RS_1^2 = \emptyset, WS_1^2 = \{c\}]$ for G_1^2 into the TTBL. The GTM also inserts G_1 into MQ[1] and MQ[2]. Since G_1 is the first global transaction, the GTM transfers G_1^1 and G_1^2 to the GTA. In case of G_2, the GTM registers the information of G_2^1 and G_2^2 into the TTBL, and inserts G_2 into MQ[1] and MQ[2]. G_2 is inserted after G_1 in MQ[1], and thus the serialization order at LDBS$_1$ is $G_1 \rightarrow G_2$. Since $WS_2^1 \cap (RS_1^1 \cup$

$WS_1^1) \neq \emptyset$, G_1 and G_2 directly conflict in LDBS_1. The same serialization order of $G_1 \rightarrow G_2$ is also applied in LDBS_2 due to the message ordering property. However, since $WS_2^2 \cap (RS_1^2 \cup WS_1^2) = \emptyset$, an additional operation of $r_2(c)$ has to be appended into the last position of G_2 to force conflicts between G_1 and G_2 (Step 2.1). Then LDBS_2 can detect a local deadlock and aborts T_3. The resulting schedule of LDBS_2 becomes as follows.

$$\text{LDBS}_2 : w_1(c)\ r_2(b)\ r_2(c),\ G_1 \rightarrow G_2$$

4 Discussion

GCC-M determines the conflict relationship of subtransactions using their data set. If the data set of a global transaction is not available before its execution, the GTM cannot define RS_k^i and WS_k^i of a subtransaction T_k^i. Then the conflict relationship can be forced by appending an additional update operation on special data item, e.g. ticket [8], stored in each LDBS. While this approach might look similar to OTM, it can outperform OTM by avoiding distributed deadlocks. This is because GCC-M with the message ordering guarantees that update operations on the tickets are executed with the same order in every LDBS.

We assume that each LDBS supports a two-phase locking. GCC-M can also support other types of local concurrency control algorithm, such as timestamp ordering and optimistic algorithm. In particular, GCC-M is well matched with the timestamp ordering. Note that the timestamp ordering ensures that transactions are executed in timestamp order. This means that if an LDBS assigns the timestamp of each subtransaction before its execution, then the MOC can satisfy the global serializability just by issuing subtransactions to the LDBS with their delivery order. Furthermore, the conflict information is not necessary and thus it is possible to omit Step 2 of GCC-M.

If an LDBS uses an optimistic concurrency control algorithm, GCC-M performs differently according to whether the LDBS can support the two-phase commitment (2PC) protocol or not. Suppose that the LDBS supports the 2PC protocol. In this case, GCC-M allows subtransactions to be executed concurrently in the LDBS. If the resulting validation order is not matched with the delivery order, corresponding global transaction cannot enter the commit phase and is aborted. On the other hand, if the LDBS does not support the 2PC protocol, GCC-M should issue subtransactions one after another into the LDBS. This is because concurrent execution of subtransactions may lead to a commitment order of subtransactions different from the delivery order, which violates the global serializability.

5 Experiments

5.1 Experiment Model

We compare the performance of GCC-M and OTM with an experiment model using the CSIM discrete-event simulation package [13]. Table 1 summarizes the experiment parameters. Much of the values are adopted from [5,6,10].

Table 1. Experiment parameters

System Parameters		
CPUSpeed	Instruction rate of node CPU	30 MIPS
NetBandwidth	Network Bandwidth	100 Mbps
NumNode	Number of nodes	3
MPL	Multiprogramming level (No. of terminals)	5 - 50
NumDisk	Number of disks per node	2 disks
MinDiskTime	Minimum disk access time	0.01 sec
MaxDiskTime	Maximum disk access time	0.03 sec
DBSize	Number of database items per node	1000
CacheHitRatio	Cache hit ratio	80%
Overhead and Transaction Parameters		
FixedMsgInst	Number of instructions per message	20000
CtlMsgSize	Size of a control message (bytes)	256
LockInst	CPU instructions for lock/unlock pair	300
PerIOInst	CPU instructions for disk I/O	5000
LTLength	Local transaction size (No. of data items)	10
STLength	Subtransaction size (No. of data items)	10
TRSizeDev	Deviation of transaction size	10%
WriteOpPct	Probability of write operation	20%

We consider an MDBS with three LDBSs located in different nodes respectively. Each node has one CPU and two disks, and each disk has a FIFO queue of I/O requests. Disk access time is drawn from a uniform distribution between 0.01 second to 0.03 second. Our network model is quite simple, acting just as a switch for routing messages between nodes. This is because our experiments assume a local area network where the actual time on the wire for messages is negligible. The network manager is implemented as a FIFO server with 100 Mbps bandwidth. The CPU cost to send or receive a message via network is modelled as a fixed number of instructions per message. The WriteOpPct parameter represents the probability of updating a data item. In our experiment model, there are two types of transaction: local transaction and global transaction. The size of a local transaction is determined by a uniform distribution between LTlength $\pm$ LTlength $\times$ TRSizeDev. A global transaction is assumed to access all LDBSs; hence, it consists of three subtransactions each of which has the same length as local transactions.

The performance metric used in the experiments is the *transaction throughput*. The transaction throughput is measured as the number of total transactions that successfully commit per second. We divide the transaction throughput into *global throughput* and the *local throughput* according to the transaction type.

To analyze the tradeoffs between OTM and GCC-M, their performance will be examined under a wide variety of database workloads. In the followings, we first describe the characteristics of each workload, and then discuss the experiment results on the workload.

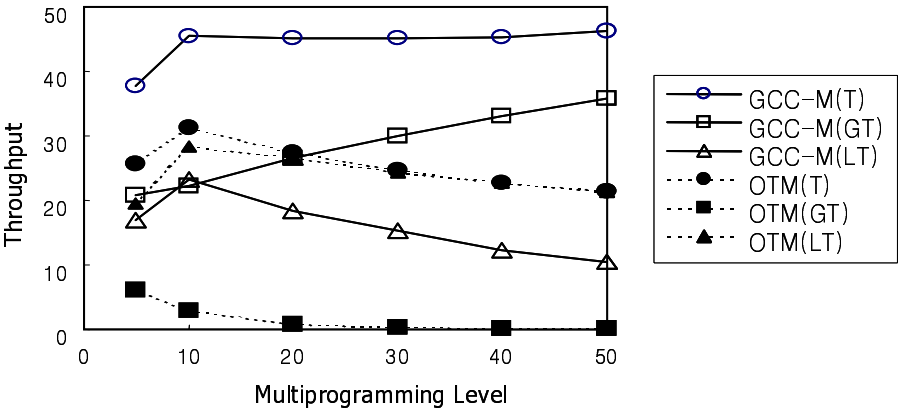


Fig. 2. High contention workload - throughput

5.2 High Contention Workload

This workload models an application where all transactions have the same data access skew and the degree of data contention is very high as a result. Most transactions access the hot set. Specifically, 80% of every transaction's accesses go to about 20% of database.

Fig. 2 shows the transaction throughput at this workload by varying the multiprogramming level (MPL). The MPL of total system is changed from 5 to 50. GCC-M outperforms OTM for every MPL. When the MPL is 50, the throughput of GCC-M is about two times compared to OTM. Note that GCC-M restricts the concurrency of global transactions by delaying subtransactions if their preceding subtransactions do not execute the conflict operations. Furthermore, since every LDBS executes global transactions with their delivery order, distributed deadlocks can be avoided. On the other hand, OTM executes more global transactions concurrently without any restrictions, and every subtransaction in an LDBS accesses a ticket. This means that OTM suffers from the high degree of data contention, and thus large number of global transactions must abort due to local and/or distributed deadlocks. This is particularly true at the large MPL where the degree of data contention is substantial. The global throughput of OTM is nearly 0 when the MPL is over 30. At this period, most of global transactions are aborted due to deadlocks.

An interesting observation is that the local throughput of OTM is higher than that of GCC-M. The reason is that the degree of data contention is mitigated and concurrency is increased relatively in case of local transactions, since most of global transactions suffer from locking delay and abort due to accessing tickets.

5.3 Partitioning Workload

This workload models an environment where transactions in an LDBS mainly access disjoint portions of the database. Specifically, each transaction (either lo-

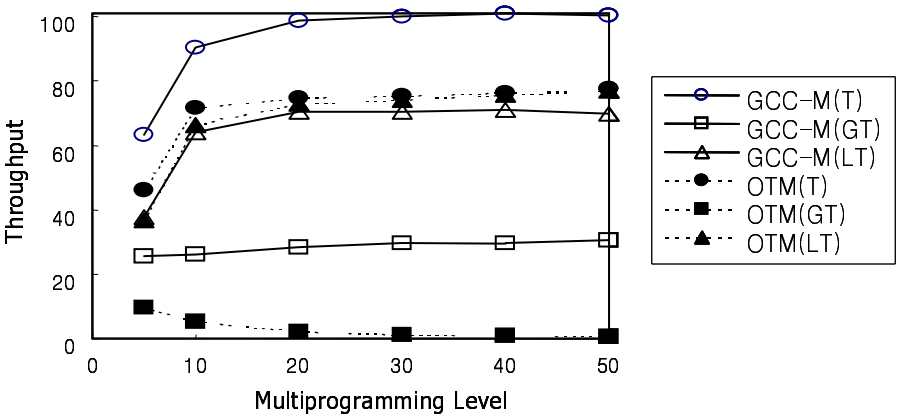


Fig. 3. Partitioning workload - throughput

cal or subtransaction) has an affinity for its own preferred region of the database, directing 80% of its accesses to that specific region. The remaining 20% go to the shared region. The shared region occupies about 20% of the database. Since there is low probability of accessing the same data item for different transactions, the degree of data contention is reduced significantly.

Fig. 3 shows the experiment results of this workload by varying the MPL. As expected, the performance of both algorithms improves dramatically due to the reduced data contention and low lock conflict ratio. In particular, the degree of performance improvement of OTM is higher than that of GCC-M. Even though the MPL increases, the performance differences are within 20%. This results from the following two reasons. First, the reduced data contention leads to lower probability of deadlock occurrence, especially between subtransactions and local transactions. This is why the local throughput of OTM improves significantly. Next, compared to the high contention workload, GCC-M may append more operations to force conflicts since the probability of direct conflicts between subtransactions must be low. This might increase the lock conflict ratio slightly between subtransactions and local transactions.

Compared to the high contention workload, the global throughput of GCC-M and OTM is not changed significantly in this workload. Note that both algorithms rely on forcing conflicts between subtransactions to ensure the global serializability. In this workload, since direct conflicts between subtransactions are not occurred frequently, more subtransactions would have additional operations to force conflicts in GCC-M. As a result, the potential improvement due to the low data contention can be offset by the processing overhead on the additional operations. The global throughput of OTM is still nearly 0 in high MPL, because the probability of occurring deadlocks due to ticket access is inherently high in OTM.

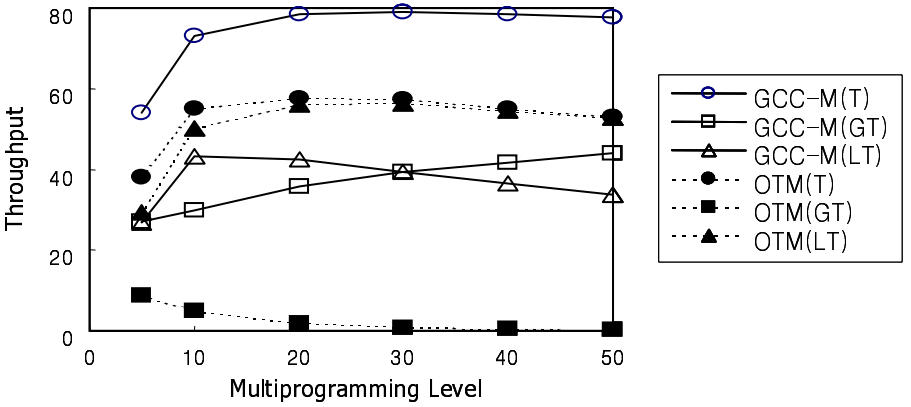


Fig. 4. Uniform workload - throughput

5.4 Uniform Workload

The last experiment was performed on the uniform workload, where all transactions in an LDBS access data items uniformly throughout the entire database. For each transaction, 90% of its operations access the entire database except the shared region, and the remaining 10% go to the shared region of the database. Similar to the partitioning workload, the shared region occupies about 20% of the database. Fig. 4 shows the experiment results of this workload by varying the MPL. The performance of all algorithms gets worse compared to the partitioning workload due to the increased probability of lock conflict. However, compared to the high contention workload, both algorithms perform better. As the MPL increases, the performance differences of GCC-M and OTM become significant and the maximum difference is about 30%.

6 Concluding Remarks

In this paper, we propose a new global concurrency control algorithm in MDBS, named GCC-M (Global Concurrency Control algorithm based on the Message ordering). GCC-M is novel in the sense that it is the first approach to adopt the idea of message ordering in group communication to the area of global concurrency control. The message ordering allows to determine easily the relative serialization order of global transactions, and thus it can make the global concurrency control algorithm be simple and can also avoid distributed deadlocks. GCC-M can exploit the potential advantages of message ordering by forcing each LDBS to preserve the delivery order of global transactions without violating local autonomy.

We have explored the performance of GCC-M under a wide variety of database workloads using the distributed database simulation model. The experiment results show that GCC-M outperforms OTM for every workload. The performance difference is significant when the degree of data contention is high.

This corresponds to the case where transactions have access skew so that part of a database is accessed more frequently or large number of transactions are executed concurrently. This feature of GCC-M is very encouraging since non-uniform database access is not rare in practice and the scale of MDBS tends to be increasing due to the development of Internet database.

References

1. Breitbart, Y., Garcia-Molina, H., Silberschatz, A.: Overview of Multidatabase Transaction Management. VLDB J. **1** (1992) 72–79
2. Breitbart, Y., Georgakopoulos, D., Rusinkiewicz, M., Silberschatz, A.: On Rigorous Transaction Scheduling. IEEE Trans. on Software Eng. **17** (1991) 954–960
3. Breitbart, Y., Silberschatz, A.: Multidatabase Update Issues. In: Proc. ACM SIGMOD (1988) 135–142
4. Chappell, D., Monson-Haefel, R.: Guaranteed Messaging with JMS. Java Developer's J. (2001)
5. Cho, H.: Cache Coherency and Concurrency Control in a Multisystem Data Sharing Environment. IEICE Trans. on Infor. Syst. **E82-D** (1999) 1042–1050
6. Cho, H., Park, J.: Maintaining Cache Coherency in a Multisystem Data Sharing Environment. J. Syst. Architecture **45** (1998) 285–303
7. FioranoMQ and Progress SonicMQ Highlights. http://www.fiorano.com
8. Georgakopoulos, D., Rusinkiewicz, M., Sheth, A.: Using Tickets to Enforce the Serializability of Multidatabase Transactions. IEEE Trans. on Knowledge and Data Eng. **6** (1994) 166–180
9. Holliday, J., Agrawal, D., Abbadi, A.: Using Multicast Communication to Reduce Deadlock in Replicated Databases. In: Proc. IEEE Symp. on Reliable Distributed Syst. (2000) 196–205.
10. Kemme, B., Alonso, G.: A New Approach to Developing and Implementing Eager Database Replication Protocols. ACM Trans. on Database Syst. **25** (2000) 333–379
11. Lee, S., Hwang, C., Lee, W.: A Uniform Approach to Global Concurrency Control and Recovery in Multidatabase Environment. In: Proc. Int. Conf. on Infor. and Knowledge Management (1997) 51–58
12. Mehrotra, S., Rastogi, R., Korth, H., Silberschatz, A.: Ensuring Consistency in Multidatabases by Preserving Two-Level Serializability. ACM Trans. on Database Syst. **23** (1998) 199–230
13. Schwetman, H.: CSIM User's Guide for use with C Revision 16. MCC (1992)
14. Getting Started with SonicMQ V4. http://www.sonicsoftware.com
15. Tanenbaum, A., van Steen, M.: Distributed Systems - Principles and Paradigms. Prentice Hall (2002)
16. Zhang, A., Elmagarmid, A.: A Theory of Global Concurrency Control in Multidatabase Systems. VLDB J. **2** (1993) 331–360

Applications of Fuzzy Data Mining Methods for Intrusion Detection Systems

Jian Guan, Da-xin Liu, and Tong Wang

College of Computer Science and Technology, Harbin Engineering University
150001 Harbin, China
{kwanjian,dxliu,twang}@0451.com

Abstract. Two data mining methods (association rule mining and frequent episode mining) have been proved to fit to the intrusion detection problem. But the normal and the intrusions in computer networks are hard to predict as the boundaries between them cannot be well defined. This prediction process may generate false alarms in many anomaly based intrusion detection systems. This paper presented a method to realize that the false alarm rate in determining intrusive activities can be reduced with fuzzy logic. A set of fuzzy rules can be used to define the normal and abnormal behavior in a computer network, and fuzzy data mining algorithms can be applied over such rules to determine when an intrusion is in progress. In this paper, we have introduced modifications of these methods that mine fuzzy association rules and fuzzy frequent episodes and have described off-line methods that utilize these fuzzy methods for anomaly detection from audit data. We describe experiments that explore their applicability for intrusion detection. Experimental results indicate that fuzzy data mining can provide effective approximate anomaly detection.

Keywords: Network Security, Intrusion Detection, Fuzzy Sets, Data Mining

1 Introduction

All over the world companies and governments are increasingly depending on their computer networks and communications, so it is becoming more and more important that protecting these systems from attack. A single intrusion of a computer network can result in the loss, unauthorized utilization or modification of large amounts of data, and cause the paralyses of normal usage of the network communications. There are numerous methods of responding to a network intrusion. In addition to intrusion protection techniques, such as user authentication and authorization, encryption, and defensive programming, intrusion detection is often used as another way to protect computer networks and systems.

Intrusion detection is a type of network security that, as the name implies, attempts to detect, identify and isolate attempts to "intrude" or make inappropriate, unauthorized use of computers. Attacks originate either via an external network connection, or from within your own organization. Target systems are usually server or workstation systems, however attackers may also focus on network devices such as hubs, routers and switches. An intrusion detection system (IDS) helps identify the fact that attacks are occurring. It may also be able to detect attacks that other security

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 706–714, 2004.
© Springer-Verlag Berlin Heidelberg 2004

components don't see and help collect forensic evidence which can be used to identify intruders. Intrusion detection systems are based on the assumption that an intruder can be detected through an examination of network traffic and of various system events such as CPU utilization, system calls, user location, and various file activities. Network sensors and system monitors convert observed events into chronologically sorted records of system activities. Called "audit trails", these records are analyzed by IDS products for unusual or suspect behavior.

There are two types of intrusion detection: misuse detection and anomaly detection. Misuse detection can be applied to the attacks that generally follow some fixed patterns. For example, three consecutive login failures are likely to be one of the important characteristics of password guessing. Misuse detection is usually constructed to examine these intrusion patterns that have been recognized and reported by experts. However, intruders do not always follow publicly known patterns to break into a computer system. They will often try to mask their illegal behavior to deceive the detection system. Anomaly detection methods are designed to counter this kind of challenge. Unlike misuse detection that is based on attack patterns, anomaly detection tries to find patterns of normal behavior, with the assumption that an intrusion will usually include some deviation from this normal behavior. Observation of this deviation will then result in an intrusion alarm.

Artificial intelligence (AI) techniques have played an important role in both misuse detection and anomaly detection [1]. AI techniques can be used for data reduction and classification tasks. For example, many intrusion detection systems have been developed as rule-based expert systems. An example is SRI's Intrusion Detection Expert System (IDES) [2]. The rules for detection can be constructed based on the knowledge of system vulnerabilities or known attack patterns. On the other hand, AI techniques also have the capability of learning inductive rules. For example, sequential patterns can be learned by a system such as the Time-based Inductive Machine (TIM) for intrusion detection [3]. Neural networks can be used to predict future intrusions after training [4]. Data mining methods have been also proposed to mine normal patterns from audit data. Lee et al., [5] describe how to use association rules and frequent episode algorithms to guide the process of audit data gathering and selection of useful features to build the classifiers.

The approaches of data mining for intrusion detection are effective. Problems are encountered, however, if one derives rules that are directly dependent on audit data. An intrusion that deviates only slightly from a pattern derived from the audit data may not be detected or a small change in normal behavior may cause a false alarm. We have addressed this problem by integrating fuzzy logic with data mining methods for intrusion detection.

Fuzzy logic is appropriate for the intrusion detection problem for two major reasons. First, many quantitative features are involved in intrusion detection. SRI's Next-generation Intrusion Detection Expert System (NIDES) categorizes security-related statistical measurements into four types: ordinal, categorical, binary categorical, and linear categorical [6]. Both ordinal and linear categorical measurements are quantitative features that can potentially be viewed as fuzzy variables. Two examples of ordinal measurements are the CPU usage time and the connection duration. An example of a linear categorical measurement is the number of different TCP/UDP services initiated by the same source host. The second motivation for using fuzzy logic to address the intrusion detection problem is that security itself includes fuzziness. Given a quantitative measurement, an interval can

be used to denote a normal value. Then, any values falling outside the interval will be considered anomalous to the same degree regardless of their distance to the interval. The same applies to values inside the interval, i.e., all will be viewed as normal to the same degree. The use of fuzziness in representing these quantitative features helps to smooth the abrupt separation of normality and abnormality and provides a measure of the degree of normality or abnormality of a particular measure. Dickerson et al., [7] developed the Fuzzy Intrusion Recognition Engine (FIRE) using fuzzy sets and fuzzy rules. FIRE uses the *Fuzzy C-Means Algorithm* developed to generate fuzzy sets for every observed feature. The fuzzy sets are then used to define fuzzy rules to detect individual attacks. FIRE does not establish any sort of model representing the quiescent state of the system, but instead relies on attack specific rules for detection.

We are combining techniques from fuzzy logic and data mining for intrusion detection system. The advantage of using fuzzy logic is that it allows one to represent concepts that could be considered to be in more than one category (or from another point of view—it allows representation of overlapping categories). In standard set theory, each element is either completely a member of a category or not a member at all. In contrast, fuzzy set theory allows partial membership in sets or categories. The second technique, data mining, is used to automatically learn patterns from large quantities of data. The integration of fuzzy logic with data mining methods helps to create more abstract and flexible patterns for intrusion detection.

The rest of the paper is organized as follows. Section 2 outlines the theory of fuzzy logic and data mining used in our framework. Section 3 briefly describes several fuzzy data mining programs, and discusses how they can be applied to discover frequent intrusion and normal activity patterns, which are the basis for building anomaly detection components. Section 4 reports the results of our experiments on building intrusion detection models using the audit data. Section 5 summaries our work, and outlines our future research plans.

2 Fuzzy Logic and Data Mining

Based on fuzzy set theory, fuzzy logic provides a powerful way to categorize a concept in an abstract way by introducing vagueness. On the other hand, data mining methods are capable of extracting patterns automatically from a large amount of data. The integration of fuzzy logic with data mining methods will help to create more abstract patterns at a higher level than at the data level. Decreasing the dependency on data will be helpful for patterns used in intrusion detection.

2.1 Fuzzy Logic

Traditionally, a standard set like $S = \{a, b, c, d, e\}$ represents the fact that every member totally belongs to the set S. However, there are many concepts that have to be expressed with some vagueness. For instance, “tall” is fuzzy in the statement of “John’s height is tall” since there is no clear boundary between “tall” and not “tall”.

Fuzzy set theory established by Lotfi Zadeh [8] is the basis of fuzzy logic. A fuzzy set is a set to which its members belong with a degree between 0 to 1. For example, $S' = \{(a\ 0), (b\ 0.3), (c\ 1), (d\ 0.5), (e\ 0)\}$ is a fuzzy set in which a, b, c, d, and e have

membership degrees in the set of *S'* of 0, 0.3, 1, 0.5, and 0 respectively. So, it is absolutely true that *a* and *e* do not belong to *S'* and *c* does belong to *S'*, but *b* and *e* are only partial members in the fuzzy set *S'*.

A fuzzy variable (also called a linguistic variable) can be used to represent these concepts associated with some vagueness. A fuzzy variable will then take a fuzzy set as a value, which is usually denoted by a fuzzy adjective. For example, "height" is a fuzzy variable and "tall" is one of its fuzzy adjectives, which can be represented by a fuzzy set.

2.2 Data Mining

Data mining generally refers to the process of extracting descriptive models from large stores of data [9]. The recent rapid development in data mining has made available a wide variety of algorithms, drawn from the fields of statistics, pattern recognition, machine learning, and databases. Several types of algorithms are particularly useful for mining audit data:

Link analysis: determines relations between fields in the database records. Correlations of system features in audit data, for example, the correlation between *command* and *argument* in the shell command history data of a user, can serve as the basis for constructing normal usage profiles. A programmer, for example, may have "emacs" highly associated with "C" files;

Sequence analysis: models sequential patterns. These algorithms can discover what time-based sequence of audit events are frequently occurring together. These frequent event patterns provide guidelines for incorporating temporal statistical measures into intrusion detection models. For example, patterns from audit data containing network-based denial-of-service (DOS) attacks suggest that several per-host and per-service measures should be included;

Classification: maps a data item into one of several predefined categories. These algorithms normally output "classifiers", for example, in the form of decision trees or rules. An ideal application in intrusion detection will be to gather sufficient "normal" and "abnormal" audit data for a user or a program, then apply a classification algorithm to learn a classifier that can label or predict new unseen audit data as belonging to the normal class or the abnormal class.

3 Intrusion Detection via Fuzzy Data Mining

Data mining methods include association rule mining and frequent episode mining.

3.1 Fuzzy Association Rules

Association rules were first developed to find correlations in transactions using retail data [9]. For example, if a customer who buys a soft drink (A) usually also buys potato chips (B), then potato chips are associated with soft drinks using the rule A _ *B*. Suppose that 25% of all customers buy both soft drinks and potato chips and that 50% of the customers who buy soft drinks also buy potato chips. Then the degree of

support for the rule is s =0.25 and the degree of confidence in the rule is c = 0.50. Agrawal and Srikant developed the fast Apriori algorithm for mining association rules [10]. The Apriori algorithm requires two thresholds of *minconfidence* (representing minimum confidence) and *minsupport* (representing minimum support). These two thresholds determine the degree of association that must hold before the rule will be mined.

In order to use the Apriori algorithm of Agrawal and Srikant [10] for mining association rules, one must partition quantitative variables into discrete categories. This gives rise to the "sharp boundary problem" in which a very small change in value causes an abrupt change in category. Kuok, Fu, and Wong [11] developed the concept of fuzzy association rules to address this problem. Their method allows a value to contribute to the support of more than one fuzzy set.

According to Kuok, Fu, and Wong's method [8], suppose we are given the complete item set $I = \{i_1, i_2, \ldots, i_m\}$ where each i_j $(1 \le j \le m)$ denotes a categorical or quantitative (fuzzy) attribute. We introduce $f(i_j)$ to represent the maximum number of categories (if i_j is categorical) or the maximum number of fuzzy sets (if i_j is fuzzy) and $m_{ij}(l,v)$ to represent the membership degree of v on the l^{th} category or fuzzy set of i_j. If i_j is categorical, $m_{ij}(l,v)=0$ or $m_{ij}(l,v)=1$. If i_j is fuzzy, $0 \le m_{ij}(l,v) \le 1$. Srikant and Agrawal [9] introduce the idea of mapping the categories (or fuzzy sets) of an attribute to a set of consecutive integers. Then an itemset X^k $(1 \le k \le m)$ can be expressed as X^k $\{item_1=c_1, item_2=c_2, \ldots, item_k=c_k\}$ where $\{X^k.item_1, X^k.item_2, \ldots, X^k.item_k\} \subseteq I$ and for all j $(1 \le j \le k)$, $1 \le c_j \le f(i_j)$.

So, given a transaction $T = \{T.i_1, T.i_2, \ldots, T.i_m\}$, $T.i_j$ $(1 \le j \le m)$ represents a value of the j^{th} attribute and can be mapped to $\{(l, m_{ij}(l, T.i_j) \mid \text{for all } l, (1 \le l \le f(i_j)))$. However, when using Kuok, Fu, and Wong's algorithm, if i_j is fuzzy,

$$\sum_{l=1}^{f(i_j)} m_{i_j}\left(l, T.i_j\right) \tag{1}$$

does not always equal to 1. We have developed a normalization process as follows:

$$m'_{i_j}\left(l, T.i_j\right) = \begin{cases} \dfrac{m_{i_j}\left(l, T.i_j\right)}{\sum_{l=1}^{f(i_j)} m_{i_j}\left(l, T.i_j\right)} & \text{If } i_j \text{ is fuzzy} \\ m_{i_j}\left(l, T.i_j\right) & \text{If } i_j \text{ is categorical} \end{cases} \tag{2}$$

Then, for an itemset X^k $\{item_1=c_1, item_2=c_2, \ldots, item_k=c_k\}$ where $1 \le k \le m$, its support contributed by T will be:

$$\prod_{j=1}^{k} m'_{X^k.item_j}\left(X^k.c_j, T.\left(X^k.item_j\right)\right) \tag{3}$$

Here we use the *product* to calculate an itemset's support because given a transaction $T = \{T.i_1, T.i_2, \ldots, T.i_m\}$ and any attribute set $\{item_1, item_2, \ldots, item_k\}$ $(1 \le k \le m)$,

$$\sum_{\forall c_j \in [1, f(item_j))] } \left(\prod_{j=1}^{k} m'_{item_j} \left(c_j, T.item_j \right) \right) = 1 \tag{4}$$

will hold. That is to say, for any item or any combination of items, the support from a transaction will be always 1.

We have modified the algorithm of Kuok, Fu, and Wong [11], by introducing a normalization factor to ensure that every transaction is counted only one time. The rest of the algorithm for fuzzy association rules is similar to the algorithm Apriori for Boolean association rules [10].

3.2 Fuzzy Frequency Episodes

Mannila and Toivonen [12] proposed an algorithm for discovering simple serial frequency episodes from event sequences based on minimal occurrences. Lee, Stolfo, and Mok [5] have applied this method to the problem of characterizing frequent temporal patterns in audit data.

The need to develop fuzzy frequency episodes comes from the involvement of quantitative attributes in an event. That is to say, given the set of event attributes $A = \{a_1, a_2, \ldots, a_m\}$, each attribute a_j $(1 \le j \le m)$ may be categorical or quantitative (fuzzy). Suppose $f(a_j)$ represents the maximum number of categories (if a_j is categorical) or the maximum number of fuzzy sets (if a_j is fuzzy), and $m_{ai}(l,v)$ represents the membership degree of v in the l^{th} category or fuzzy set of a_j. If a_j is categorical, $m_{ai}(l,v)=0$ or $m_{ai}(l,v)=1$. If a_j is fuzzy, $0 \le m_{ai}(l,v) \le 1$. Similarly, for an event attribute, its categories or fuzzy sets can be mapped to consecutive integers. Then an event variable e^k can be expressed as $e^k\{attr_1=c_1, attr_2=c_2, \ldots, attr_k=c_k\}$ where $\{e^k.attr_1, e^k.attr_2, \ldots, e^k.attr_k\} \subseteq \boldsymbol{A}$ and for all j $(1 \le j \le k)$, $1 \le c_j \le f(a_j)$. We define two event variables $e^p\{attr_1=c_1, attr_2=c_2, \ldots, attr_p=c_p\}$ and $e^q\{attr'_1=c_1, attr'_2=c_2, \ldots, attr'_q=c_q\}$ as *homogeneous*, if $\{e^p.attr_1, e^p.attr_2, \ldots, e^p.attr_p\} = \{e^q.attr'_1, e^q.attr'_2, \ldots, e^q.attr'_q\}$, which also indicates that $p = q$. It is obvious that an event variable is homogeneous to itself.

So, given an event $E = \{E.a_1, E.a_2, \ldots, E.a_m\}$, $E.a_j$ $(1 \le j \le m)$ represents a value of the j^{th} attribute and can be mapped to $\{(l, m_{aj}(l, E.a_j) \mid \text{for all } l, (1 \le l \le f(a_j)))\}$. However, if a_j is fuzzy,

$$\sum_{l=1}^{f(a_j)} m_{a_j} \left(l, E.a_j \right) \tag{5}$$

does not always equal to 1. A normalization process is used as follows:

$$m'_{a_j} \left(l, E.a_j \right) = \begin{cases} \dfrac{m_{a_j} \left(l, E.a_j \right)}{\sum_{l=1}^{f(a_j)} m_{a_j} \left(l, E.a_j \right)} & \text{If } a_j \text{ is fuzzy} \\ m_{a_j} \left(l, E.a_j \right) & \text{If } a_j \text{ is categorical} \end{cases} \tag{6}$$

Then, for an event variable $e^k\{attr_1=c_1, attr_2=c_2, \ldots, attr_k=c_k\}$ where $1 \le k \le m$, its occurrence in E is no longer counted as either 0 or 1. Instead, it is defined as:

$$occurrence\left(e^k, E\right) = \prod_{j=1}^{k} m'_{e^k.attr_j}\left(e^k.c_j, E.\left(e^k.attr_j\right)\right) \tag{7}$$

And the minimal occurrence of an episode is the *product* of the occurrences of its event variables.

We have modified the method of Mannila and Toivonen [12] to mine to fuzzy frequency episodes. In Mannila and Toivonen's method [12], an event is characterized by a set of attributes at a point in time. An episode *P(e1,e2, …, ek)* is a sequence of events that occurs within a time window [*t,t*']. The episode is minimal if there is no occurrence of the sequence in a subinterval of the time interval. Given a threshold of *window* (representing timestamp bounds), the frequency of *P(e1,e2, …, ek)* in an event sequence ***S*** is the total number of its minimal occurrences in any interval smaller than *window*. So, given another threshold *minfrequency* (representing minimum frequency), an episode *P(e1,e2, …, ek)* is called frequent, if *frequency(P)/n e minfrequency*.

The need to develop fuzzy frequency episodes comes from the involvement of quantitative attributes in an event. Other than the difference in calculating the frequency (or minimal occurrence) of an episode, our algorithm is similar to Mannila and Toivonen's algorithm [11] for mining frequency episodes. An example of a fuzzy frequency episode rule mined by our system is given below:

{E1: PN=*LOW*, E2: PN=*MEDIUM*} -> {E3: PN=*MEDIUM*}, c = 0.854, s = 0.108, w = 10 seconds

where E1, E2, and E3 are events that occur in that order and PN is the number of distinct destination ports within a 2 second period.

4 Experiments and Results

This section reports upon program performance, as well as analyzing how effective various attempts at applying fuzzy data ming have been. To ensure fair comparisons, all analysis will be performed under the same testing environment, which is:

- 2.4 GHz Pentium IV-powered PC with 256Mb DDR RAM;
- Redhat Linux 8.0 Personal operating system.

The experiments was designed to investigate the applicability of fuzzy association rules and fuzzy frequency episodes for anomaly detection. The experimental data come from the Department of Computer Science at Mississippi State University. One of the servers in the Department of Computer Science at Mississippi State University has been monitored and its real-time network traffic data has been collected by *tcpdump*. [5] suggested several quantitative features of network traffic that they feel can be used for intrusion detection. Based on their suggestions, a program has been written to extract the following four temporal statistical measurements from the network traffic data:

SN – the number of SYN flags appearing in TCP packet headers during last 2 seconds;

FN – the number of FIN flags appearing in TCP packet headers during last 2 seconds;

RN – the number of RST flags appearing in TCP packet headers during last 2 seconds;

PN – the number of different destination ports during last 2 seconds.

Normal patterns (represented by fuzzy association rules and fuzzy episode rules) are first established by mining the training data. An example of a fuzzy association rule mined from the training data is: {SN = *LOW*, FN = *LOW*} -> {RN = *LOW*}, 0.924, 0.49. This means the pattern {SN = *LOW*, FN = *LOW*, RN = *LOW*} occurred in 49% of the training cases. In addition, when {SN = *LOW*, FN = *LOW*} occurs, there will be 92.4% probability that {RN = *LOW*} will also occur. An example of a fuzzy episode rule is: {PN = *LOW*, PN = *MEDIUM*} -> {PN = *MEDIUM*}, 0.854, 0.108, 10 seconds. This means that with a *window* threshold of 10 seconds, the frequency of the serial episode {PN = *LOW*, PN = *MEDIUM*, PN = *MEDIUM*} is 10.8% and when {PN = *LOW*, PN = *MEDIUM*} occurs, {PN = *MEDIUM*} will follow with an 85.4% probability.

Then for each test case, new patterns were mined using the same algorithms and the same parameters. These new patterns were then compared to the normal patterns created from the training data. If they are similar enough, no intrusion is detected; otherwise, an anomaly will be alarmed.

The similarity function proposed in [5] used a user-defined threshold, e.g., 5%. Given two rules with the same LHS and RHS, if both their confidences and their supports are within 5% of each other, these two rules are considered similar. This approach exhibits [11] sharp boundary problem. For example, given a rule R which represents a normal pattern and two test rules R' and R'', if both R' and R'' fall inside the threshold, there will be no measurement of the difference between the similarity of R and R' and the similarity of R and R''. Likewise, when both R' and R'' fall outside the threshold, there is no measure of their dissimilarities with R.

The purpose of the first experiment in this set was to determine the amount of training data (duration) needed to demonstrate differences in behavior for different time periods. In this experiment, training sets of different duration (all from the same time period, i.e., afternoon) were used to mine fuzzy association rules (see Table 1 for a more detailed description of the data). The similarity of each set of rules derived from training data of different duration was compared to test data for different time periods.

The results show that the fuzzy association rules derived from test data for the same time of the day as the training data were very similar to the rules derived from the training data. Rules derived from evening data were less similar and rules derived from late night data were the least similar. This confirms the hypothesis that fuzzy association rules are able to distinguish different behavior.

5 Conclusion and Future Work

Intrusion detection is an important but complex task for a computer system. Many AI techniques have been widely used in intrusion detection systems. Data mining methods are capable of extracting patterns automatically and adaptively from a large amount of data. Association rules and frequency episodes have been used to mine

training data to established normal patterns for anomaly detection. However, these patterns are usually at the data level, with the result that normal behavior with a small variance may not match a pattern and will be considered anomalous. In addition, an actual intrusion with a small deviation may match the normal patterns and thus not be detected. We have demonstrated that the integration of fuzzy logic with association rules and frequency episodes generates more abstract and flexible patterns for anomaly detection.

We are currently building intrusion detection components, the decision module, additional machine learning components, and a graphical user interface for the system. Also under investigation, are possible solutions to the problem of dealing with "drift" in normal behavior. We plan to extend this system to operate in a high performance cluster computing environment.

References

1. Frank J. Artificial intelligence and intrusion detection: Current and future directions. In Proceedings of the 17th national computer security conference held in October, 1994
2. Lunt T., Jagannathan R. A prototype real-time intrusion-detection expert system. In Proceedings of 1988 IEEE computer society symposium on research in security and privacy held in Oakland, California, April 18-21, 1988, by IEEE Computer Society, 59–66. Los Alamitos, CA: IEEE Computer Society Press
3. Teng Chen H., Lu K., S.. Adaptive real-time anomaly detection using inductively generated sequential patterns. In Proceedings of 1990 IEEE computer society symposium on research in security and privacy held in Oakland, California, May 7-9, 1990, by IEEE Computer Society, 278–84. Los Alamitos, CA: IEEE Computer Society Press
4. Debar H., Becker M., Siboni D.. A neural network component for an intrusion detection system. In Proceedings of 1992 IEEE computer society symposium on research in security and privacy held in Oakland, California, May 4-6, 1992, by IEEE Computer Society, 240–50. Los Alamitos, CA: IEEE Computer Society Press.
5. Lee W., Stolfo S., Mok K.. A data mining framework for building intrusion detection models. In Proceedings of the 1999 IEEE Symposium on Security and Privacy, May 1999: 120–132
6. Lunt T.. Detecting intruders in computer systems. In Proceedings of 1993 conference on auditing and computer technology
7. Dickerson J E., Juslin J, Loulousoula O, Dickerson J A.. Fuzzy Intrusion Detection. IFSA World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference, 2001
8. Zadeh L. A. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3
9. Han J, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers. 2001
10. Agrawal R., Srikant R.. Fast algorithms for mining association rules. In Proceedings of the 20th international conference on very large databases held in Santiago, Chile, September 12-15, 1994, 487–99. San Francisco, CA: Morgan Kaufmann
11. Kuok C., Fu A., Wong M.. Mining fuzzy association rules in databases. SIGMOD Record 17(1): 41–46
12. Mannila, H., Toivonen H.. Discovering generalized episodes using minimal occurrences. In Proceedings of the second international conference on knowledge discovery and data mining held in Portland, Oregon, August, 1996, by AAAI Press, 146–151

Pseudo-Random Binary Sequences Synchronizer Based on Neural Networks

Jan Borgosz and Bogusław Cyganek

AGH - University of Science and Technology
30-059 Krakow, Poland
{borgosz,cyganek}@agh.edu.pl

Abstract. Modern telecommunication market grows rapidly. Regular user of the telecommunication networks gets higher data rates and higher quality of services every year. Inherent element of this rapid progress of the services is the need to develop faster and better devices for network testing. Very important role in the test environment play Pseudo-Random Binary Sequences (PRBS) generators and synchronizers. Their functionality is described by the logical equations from the years, so it seems to be useless to make new research about them. However, experiments described in this publication show something just opposite. Authors of this paper have improved design of the PRBS synchronizer with neural network and new protocol. Proposed implementation has ability to get faster in the synchronization state and it is more resistant to the transmission errors. Finally, neural network synchronizer have overall better parameters than classic solutions, what will be subject of presented research.

1 Introduction

Telecommunication testers are devices built up with blocks known from the years. Only one parameter changes continuously – the clock frequency. Other things, like algorithms or test protocols remain unchanged. However, some improvements may be still done, what will be proved in this paper [3].

PRBS generators and receivers are commonly used for the Bit Error Ratio (BER) tests [5], which allow diagnosing different problems in the telecommunication networks, like:

1. Protocol errors;
2. Containers inconsistency;
3. Mapping and framing troubles;
4. Symbol interference;
5. Jitter transfer functions.

Generator of PRBS is circuit which works in one of the two states:

1. Idle;
2. Generation of the PRBS.

However, each PRBS receiver has four states of the work:

1. Idle;
2. Lost of synchronization;

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 715–721, 2004.
© Springer-Verlag Berlin Heidelberg 2004

3. Getting synchronization;
4. Synchronization.

State called *Idle* is state in which process of the generation or receive is paused – for example during the transmission Synchronous Overhead (SOH) in SDH systems.

For the receiver, pass between state numbers 2 and 3 is most critical point. The some situation takes place for pass between state 3 and 1. Both passes are different in their nature.

In the second case, synchronized receiver works independently from coming sequence. If total amount of the errors exceed given threshold, than receiver stops and passes to the state 1.

In the case of the state change from the 2 to 3, receiver uses previously received and buffered sequence to predict next coming value. If predicted value is equal to the received, than PRBS receiver tries to predict next value, if not, PRBS receiver reloads predictor buffer.

As may be easily seen, if during getting synchronization receiver gets value with an error, than predicted value will be corrupted too, so synchronization process will be longer [2].

In this paper, a novel way of the construction of the PRBS receiver based on the neural network is shown. Obtained results show that such approach is more resistible to the transmission errors than the classic one and has better parameters, what will be presented in the next paragraphs.

2 PRBS Definitions

Pseudo Random Bit Sequences are defined by the specially selected logic polynomials. Hardware implementation is possible with the negated *Exclusive Or* (XOR) gate and registered buffer [2][5].

2.1 Classic Implementation

Classic hardware implementation is depicted in Fig. 1. There are two main building blocks:

1. Buffer with *m+1* elements, indexed from 0;
2. Negated *XOR* gate.

Whole circuit is synchronous, what means, that during each clock cycle:

1. New value is generated and written to the latch with index 0.
2. Buffer is shifted to the left; so previous value at the *m*'th register is the value of the output. Freshly generated value is latched at position 1.

In the case of the receiver, there is additional buffer with length *m+1*. Values collected in this buffer are loaded to the buffer connected with XOR during getting synchronization [5][2].

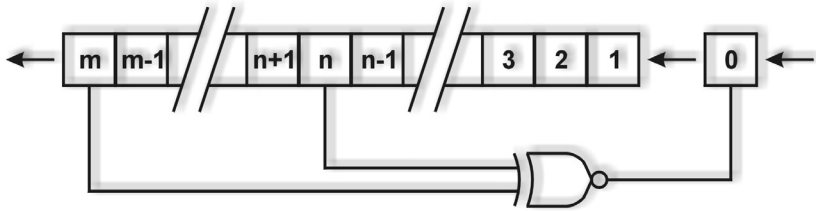


Fig. 1. PRBS generator structure

Table 1. Most popular Pseudo Random Bit Sequences used in the telecommunication tests

Name of the sequence	Period of the sequence	Polynomial	Value of the *m* coefficient	Value of the *n* coefficient
PRBS 9	511 bits	$x^9 + x^5$	9	5
PRBS 11	2047 bits	$x^{11} + x^9$	11	9
PRBS 15	32767 bits	$x^{15} + x^{14}$	15	14
PRBS 20	1048575 bits	$x^{20} + x^3$	20	3
PRBS 23	8388607 bits	$x^{23} + x^{18}$	23	18

2.2 Polynomials

There are several types of the PRB sequences [5]. Most important are presented in Table 1.

As maybe seen, period of the sequence depends on the order of the polynomial – higher order means longer period. Also the last element in the buffer is the first input to the XOR. Second element used for *XOR* calculation depends on the polynomial type. Polynomials with shorter period are rather used for the PDH systems, when polynomials with longer period are used for the SDH systems.

3 Synchronizer Based on Neural Networks

Neural networks are very powerful tools for the applications, which do require pattern recognition or function approximation in noisy environment. PRBS synchronizer will be the next proof this thesis. As may be seen in the next chapters, neural networks allow faster synchronizing, even in causes, when classic solution is unable to do it.

All experiments presented in this paper were done with Neural Networks Toolbox Packed distributed as an extension to the standard Matlab application [1].

3.1 Structure

After many experiments, authors decide to use feedforward network, connected with shift register. Other option was to use self standing Elman network. This solution has been rejected due to troubles with network learning, and pure results. Structure of the used network in notation of the MathWorks [1] is presented in Fig. 2.

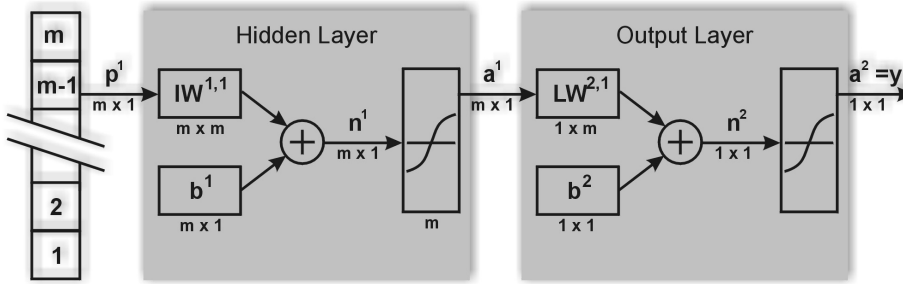


Fig. 2. Neural network structure

IW denotes weight matrix for the input layer, *LW* denotes weight matrix for the output layer, *b* is used for bias vectors, and *a* notifies output vectors. Superscripts code number of the layer, what allows distinguishing between the weight matrices, output vectors, etc. Size of each element is in the label of its graphical representation. Network depicted in Fig. 2 may be described by two equations:

$$a^1 = \log sig\left(IW^{1,1} \cdot p^1 + b^1\right) \tag{1}$$

$$y = a^2 = \log sig\left(LW^{2,1} \cdot a^1 + b^2\right) \tag{2}$$

where *logsig* is Log-Sigmoid Transfer Function [8][7][6]. Structure of the complete receiver / synchronizer is presented in Fig. 3.

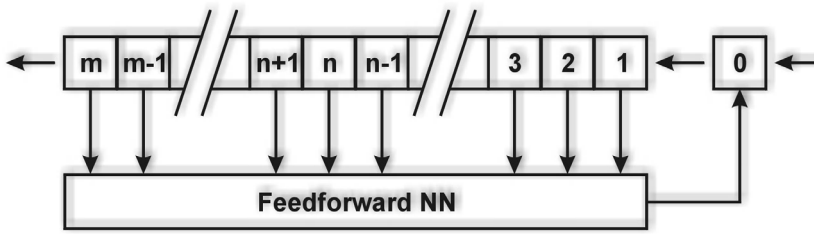


Fig. 3. Neural network based PRBS generator / receiver structure

3.2 New Protocol

In the standard synchronization protocol, errors are not allowable during first *N* steps of the synchronization, where *N* depends on the PRBS type:

$$N = 2 \cdot PRBSType \tag{3}$$

I.e. for PRBS11 *PRBSType* is equal to 11. It means, that even one error during 22 steps will cause restart of the synchronization procedure. In opposition to such approach, authors propose new synchronization protocol. This proposal assumes possibility of the occurring one error during whole synchronization procedure (*N* first steps). This change may be done due to neural synchronizer properties.

4 Experiments

In this section authors present results of their experiments with neural network synchronizer for the PRBS receivers. Ways of the learning, verification and final results are described.

Table 2. Learning results for Levenberg-Marquardt algorithm

Name of the sequence	Length of the learning sequence	Length of the test sequence	Learning error (without rounding to 0 or 1)	Error (with rounding to 0 or 1)
PRBS 9	256 bits	255 bits	$1.54*10^{-7}$	0
PRBS 11	1024 bits	1023 bits	$0.67*10^{-7}$	0
PRBS 15	16384 bits	16383 bits	$0.71*10^{-7}$	0
PRBS 20	16384 bits	16384 bits	$0.82*10^{-7}$	0
PRBS 23	16384 bits	16384 bits	$0.80*10^{-7}$	0

Table 3. Results for scenario when error occurs in the first step of the synchronization

Error Ratio	Classic synchronization mechanism reaction	Neural network based synchronization mechanism reaction
One error at the first step of the synchronization	Unable to synchronize, due to nature of the synchronizer	Able to synchronize, due to NN advantages
Two or more errors up to threshold *T*	Unable to synchronize, due to nature of the synchronizer	Able to synchronize, due to NN advantages
Count of errors exceed threshold *T* at the first step of the synchronization	Unable to synchronize, due to nature of the synchronizer	Unable to synchronize, NN is unable to recognize pattern and predict next value

Table 4. Results for scenario when error occurs in the next steps of the synchronization

Error Ratio	Classic synchronization mechanism reaction	Neural network based synchronization mechanism reaction
One error for the period of the synchronization (N)	Unable to synchronize due to the high level protocol	Able to synchronize, after high level protocol modification
Two or more errors for the period (N)	Unable to synchronize due to the high level protocol	Unable to synchronize due to the high level protocol

4.1 Neural Network Learning

During our experiments we use with success two learning algorithms: Levenberg-Marquardt and Resilient Backpropagation. First is used generally on function approximation problems, for small networks that contain up to a few hundred weights and is able to obtain lower mean square errors [6][1]. Second is faster on pattern recognition problems. Both algorithms have comparable convergence with little advantage Levenberg-Marquardt [4]. Results of the training are presented in Table 2. Please note, that learning sequence was relatively shorter for the two last cases.

4.2 Tests with NN Synchronizer

All tests were performed with Matlab and Simulink environments. Experiments were done for PRBS 9, 11, 15, 20 and 23. In our experiments we can distinguish two different scenarios:

1) Error or sequence of errors occurs at the moment of the start of the synchronization process (error is injected into buffer);
2) Error occurs in the next steps of the synchronization process.

Two tables (Table 3 and Table 4) summarize results of the experiments for both scenarios and different experiment configurations. It is easy to see, that in the all cases, proposed method gives results better or comparable, but never worse than classic solution.

Threshold T depends on PRBS type (longer PRBS, higher T) and learning sequence (longer sequence, higher T), i.e. for PRBS9 experimentally found T was 4.

We want to remind reader of fact that new protocol assumes possibility of the occurring one error during N steps of waiting for synchronization. In classic solution any error during N steps causes restart of the algorithm.

5 Conclusions

This paper describes new design of the PRBS synchronizer with the neural network element with changed algorithm of the synchronization due to its advantages. Presented implementation has better parameters than the classic solutions, especially for the corrupted data sequences. Very good results, which were obtained during experiments, make the author interested in implementing this solution in FPGA, what will be a further work.

References

1. Demuth H., Beale M.: Neural Network Toolbox, Mathworks (2003)
2. Feher and Engineers of Hewlett-Packard: Telecommunication Measurements Analysis and Instrumentation. Hewlett-Packard (1991)
3. Glover I. A., Grant P.M.: Digital Communications. Prentience Hall (1991)

4. Haykin, S.: Neural Networks. A Comprehensive Foundation. Prentice Hall (1999)
5. ITU-T: Specification O.150 - Digital test patterns for performance measurements on digital transmission equipment, ITU-T (1992)
6. Osowski S.: Sieci neuronowe w ujęciu algorytmicznym (in Polish). Wydawnictwa Naukowo – Techniczne (1996)
7. Rutkowska D., Piliński M., Rutkowski L. (1997) Sieci neuronowe, algorytmy genetyczne i systemy rozmyte (in Polish). Wydawnictwo Naukowe PWN, ISBN 83-01-12304-4.
8. Tadeusiewicz R.: Neural Networks (in Polish). Akademicka Oficyna Wydawnicza (1993)

Calculation of the Square Matrix Determinant: Computational Aspects and Alternative Algorithms

Antonio Annibali and Francesco Bellini

University of Rome 'La Sapienza'
Faculty of Economics
Department of Mathematics for Economic, Financial and Insurance Decisions
Via del Castro Laurenziano, 9 (00161) Roma / Italia
{aannib,fbellini}@scec.eco.uniroma1.it

Abstract. The calculation of a square matrix determinant is a typical matrix algebra operation which, if applied to big matrixes, asks for complex calculations.
There are different algorithms for the determinant calculation, each one with different features under the aesthetic, functional and efficiency point of view.
Besides two traditional methods such as

- the algorithmic definition,
- the first Laplace's theorem,

during this work will be shown another method based on the primitive function – provided by the APL environment – that performs the calculation of a non singular square matrix inverse.
Peculiar feature of some of the used algorithms is to be structurally recursive, but it is already possible to use the APL reduction operator – that plays as a valid algorithmic alternative – without the traditional lacks in the memory management that normally characterize the recursive procedures.

1 Algorithmic Definition

Given a square matrix A of *n* order, $\det(A)$ is a number that is determined by means of the specific rule

$$\det(A) = \sum_{k=1}^{n!} (-1)^{clas(\{p_1^{(k)}, p_2^{(k)}, ..., p_n^{(k)}\})} \prod_{j=1}^{n} a_{j, p_j^{(k)}}$$

where the summation is taken over by the whole of permutations of the natural numbers 1,2,...,n, and, consequently, contains n! summands, being

$$\{p_1^{(k)}, p_2^{(k)}, ..., p_n^{(k)}\} \qquad k = 1, 2, ..., n!$$

the different n! permutations and

$$clas(\{p_1^{(k)}, p_2^{(k)}, ..., p_n^{(k)}\}) \qquad k = 1, 2, ..., n!$$

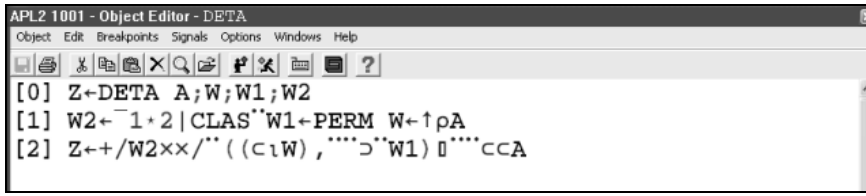
the correspondent classes according to the base permutation: 1,2,…,n.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 722–728, 2004.
© Springer-Verlag Berlin Heidelberg 2004

APL2 functions for the algorithmic calculation of the determinant are:

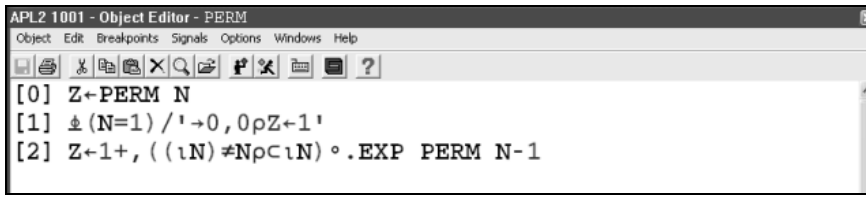
- **DETA** monadic function:

  Determinant calculation



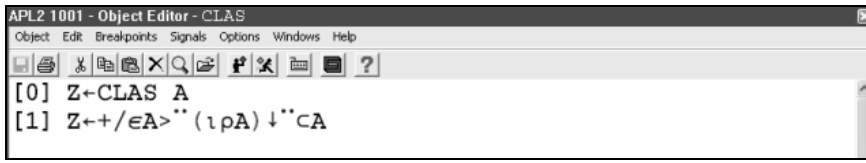
- **PERM** monadic recursive function:

  Calculation of n! permutations of first n natural numbers



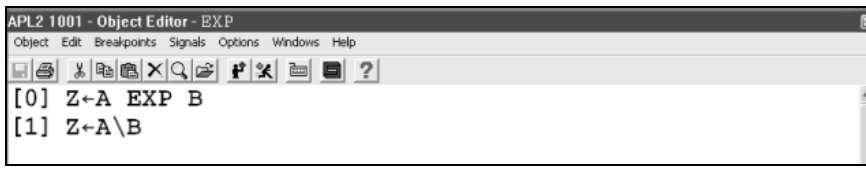
- **CLAS** monadic function:

  Calculation of permutation class



- **EXP** dyadic function:

  Service function



2 First Laplace's Theorem

Given a square matrix A of *n* order, $\det(A)$ is a number given by the sum of the n products of a line (row or column) elements for the correspondent algebraic complements

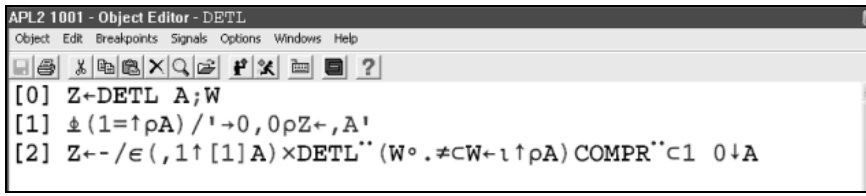
$$\det(A) = \sum_{k=1}^{n} a_{h,k} \cdot (-1)^{h+k} \cdot \det(A_{h;k}) \qquad h = 1,2,...,n$$

$$\det(A) = \sum_{h=1}^{n} a_{h,k} \cdot (-1)^{h+k} \cdot \det(A_{h;k}) \qquad k = 1,2,...,n$$

APL2 functions for the determinant calculation by means of the first Laplace's Theorem are:

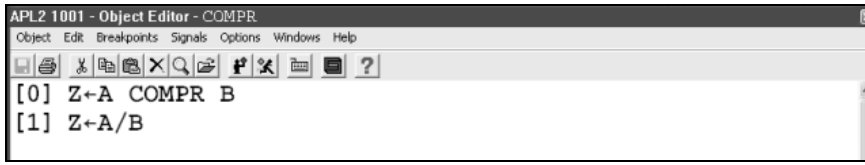
- **DETL** monadic recursive function:

  Determinant calculation



- **COMPR** dyadic function:

  Service function



3 Inverse Matrix Algorithm

Given a square matrix A (not singular) of *n* order, the generic element of the inverse matrix A^{-1} is given by the ratio between the algebraic complement of the original matrix element and the determinant of the above matrix

$$a^{-1}_{h,k} = \frac{(-1)^{h+k} \cdot \det(A_{h;k})}{\det(A)} \qquad h,k = 1,2,...,n$$

In particular, if h = k = 1 and $A_{1,1}$ is not singular, results

$$a_{1,1}^{-1} = \frac{\det(A_{1;1})}{\det(A)} \quad , \quad \det(A) = \frac{\det(A_{1;1})}{a_{1,1}^{-1}}$$

Considering that the element $a_{1,1}^{-1}$ is given by the primitive function `u'` the last formula shows that the problem of the determinant calculation of the A matrix (of *n* order) can be solved by calculating the determinant of the matrix $A_{1;1}$ (of *n-1* order).

In the same way (with $A_{1,2;1,2}$ not singular) results

$$\det(A_{1;1}) = \frac{\det(A_{1,2;1,2})}{{}^{(1)}a_{1,1}^{-1}}$$

being $A_{1,2;1,2}$ the matrix obtained from the original matrix by deleting the elements of the first two rows and columns and being ${}^{(1)}a_{1,1}^{-1}$ the element of the first row and first column of the matrix $A_{1;1}^{-1}$ that is the inverse of the matrix $A_{1;1}$.

By indicating with ${}^{(0)}a_{1,1}^{-1}$ the element $a_{1,1}^{-1}$, results

$$\det(A) = \frac{\det(A_{1,2;1,2})}{{}^{(0)}a_{1,1}^{-1} \cdot {}^{(1)}a_{1,1}^{-1}}$$

and in general (with $A_{1,2,\ldots,k;1,2,\ldots,k}$ non singular)

$$\det(A_{1,2,\ldots,k-1;1,2,\ldots,k-1}) = \frac{\det(A_{1,2,\ldots,k;1,2,\ldots,k})}{{}^{(k-1)}a_{1,1}^{-1}} \qquad k = 1,2,\ldots,n-1$$

$$\det(A) = \frac{\det(A_{1,2,\ldots,k;1,2,\ldots,k})}{\prod_{j=0}^{k-1} {}^{(j)}a_{1,1}^{-1}} \qquad k = 1,2,\ldots,n-1$$

and finally

$$\det(A_{1,2,\ldots,n-1;1,2,\ldots,n-1}) = \frac{1}{{}^{(n-1)}a_{1,1}^{-1}} \quad , \quad \det(A) = \frac{1}{\prod_{j=0}^{n-1} {}^{(j)}a_{1,1}^{-1}}$$

The determinant of a not singular matrix (under the above mentioned conditions) can be obtained as the product of the reciprocal of the elements ${}^{(j)}a_{1,1}^{-1}$ of the inverse matrices obtained by deleting the first j rows and the first j columns of the original matrix, being *j=0,1,…,n-1*.

APL2 function and operators for the determinant calculation (also by means of the reductive mode) with the (inverse) algorithm of the inverse matrix are:

- **DET** monadic recursive function:

  Determinant calculation

```
APL2 1001 - Object Editor - DET
Object Edit Breakpoints Signals Options Windows Help
[0] Z←DET MAT
[1] ⍎(0=↑⍴MAT)/'→0,0⍴Z←1'
[2] Z←(DET 1 1↓MAT)÷↑,⊞MAT
```

- **DFUNZ** dyadic function:

  Base iterative step function

```
APL2 1001 - Object Editor - DFUNZ
Object Edit Breakpoints Signals Options Windows Help
[0] Z←W DFUNZ DATIM;MAT;DETR
[1] (MAT DETR)←DATIM
[2] Z←(1 1↓MAT)(DETR÷↑,⊞MAT)
```

- **DET1** monadic reductive function:

  Determinant calculation

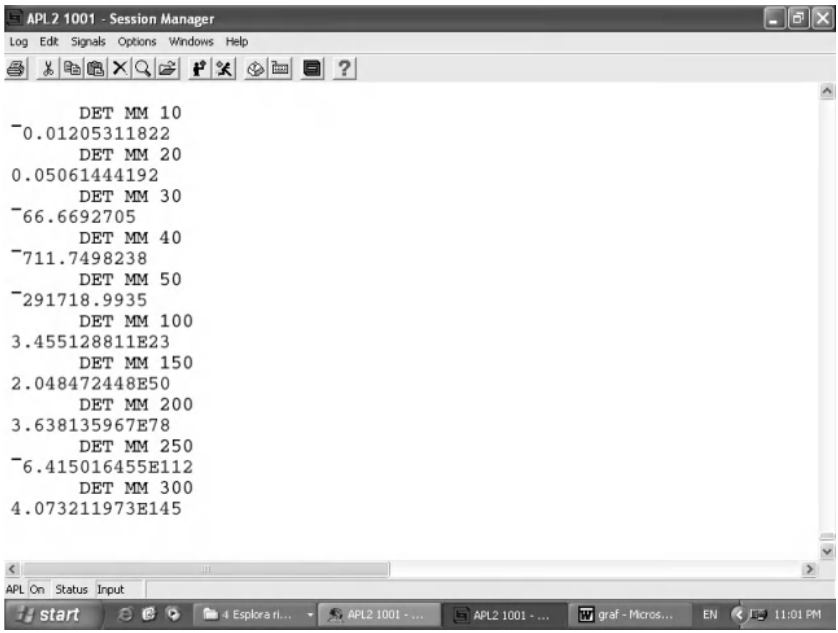
```
APL2 1001 - Object Editor - DET1
Object Edit Breakpoints Signals Options Windows Help
[0] Z←DET1 DATIM
[1] Z←2⊃⍣¨DFUNZ/(1+↑⍴1⊃DATIM)/⊂DATIM
```

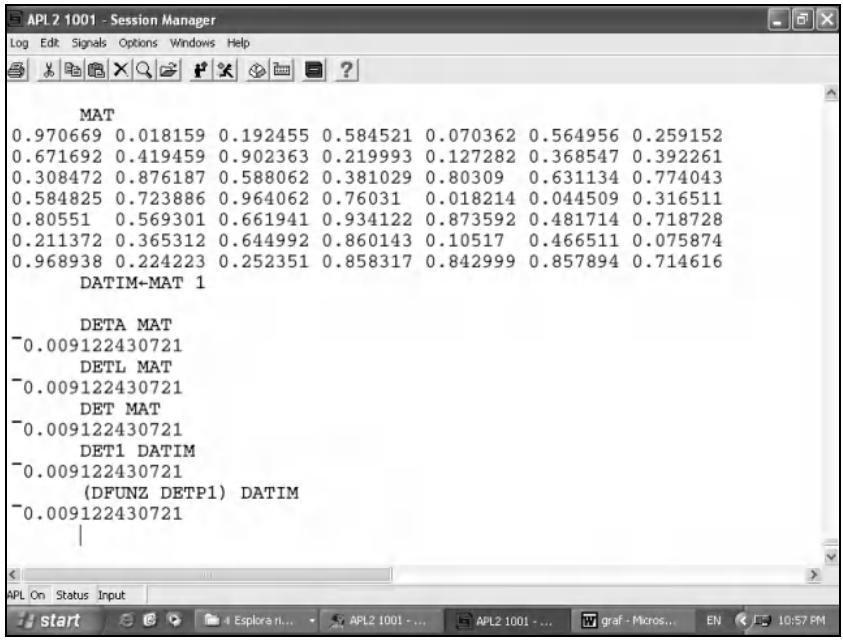
- **DETP1** monadic reductive operator:

  Determinant calculation

```
APL2 1001 - Object Editor - DETP1
Object Edit Breakpoints Signals Options Windows Help
[0] Z←(DFUNZ DETP1)DATIM
[1] Z←2⊃⍣¨DFUNZ/(1+↑⍴1⊃DATIM)/⊂DATIM
```

4 Numeric Examples[1]





[1] Where **MM N** is an N order square matrix.

References

1. J.A. Brown, S. Pakin, R.P. Polivka: *APL2 at a glance*, Prentice Hall
2. N.D. Thompson, R.P. Polivka: *Apl2 in depth*, Springer Verlag
3. D.E. Knuth: *The art of computer programming*, Addison Wesley
4. I.B.M.: *Apl2 Programming – Language Reference*

Differential Algebraic Method for Aberration Analysis of Electron Optical Systems

Min Cheng[1], Yilong Lu[1], and Zhenhua Yao[2]

[1] Division of Communication Engineering, Nanyang Technological University, Singapore 639798
[2] Singapore-MIT Alliance, National University of Singapore, Singapore 117576

Abstract. Differential algebraic method is a powerful technique in computer numerical analysis. It presents a straightforward method for computing arbitrary order derivatives of functions with extreme high accuracy limited only by the machine error. When applied to nonlinear dynamics systems, the arbitrary high order transfer properties of the system can be derived directly. In this paper, the principle of differential algebraic method is applied to calculate high order aberrations of electron optical systems. As an example, an electrostatic lens with an analytical expression has been calculated using this method. Relative errors of the Gaussian properties and spherical aberration coefficient of the lens compared with the analytic solutions are of the order 10^{-11} or smaller. It is proved that differential algebraic aberration method is very helpful with high accuracy for high order aberration analysis and computation of electron optical systems.

Keywords: Differential algebra; Electron optical systems; Aberration analysis

1 Introduction

With the increasing development of high definition display devices and electron beam lithography techniques, it has become of great importance to improve the aberration performance of high-resolution electron optical systems. Then it is necessary to investigate higher order aberrations of the systems. Various theoretical tools have been developed to deal with the high order aberration analysis and correction, such as approximately analytical method [1], canonical theory [2], and Lie algebra method [3]. These methods simplify the derivation of high order aberrations, but they have little advantage in numerical calculation and computer programming. What is more, the complexity of the expressions of aberration coefficients increases dramatically with the order of aberrations. In contrast, differential algebra method provides a powerful technique for high order aberration analysis and numerical calculation of electron optical systems.

In this paper, differential algebraic aberration method of electron optical systems is presented and is applied to describe Gaussian optical properties and high order aberrations. As an example, the Gaussian properties and the third order

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 729–735, 2004.
© Springer-Verlag Berlin Heidelberg 2004

aberrations have been calculated for *Schiske*'s model electrostatic lens, which is an extensively studied model [4]. Relative errors of the Gaussian properties compared with the analytic solutions are on the scale of 10^{-11} or small. It is shown that differential algebraic method is very effective for high order aberration analysis of electron optical systems.

2 Principle of Differential Algebra

Differential algebra is a subset of a generalization of the real numbers first introduced in the theory of nonstandard analysis [5]. In this generalization, infinitely small quantities and infinitely large quantities are united with the real numbers in a consistent way. There are also some connections to the theories of formal power series [6] and automated differentiation [7].

2.1 Arithmetic Operations in Differential Algebra

We define $N(n, v)$ to be the number of monomials in v variables through order n, that is, $N(n, v) = (n + v)!/(n!v!) = C(n + v, v)$. All these monomials are arranged in a certain manner order by order and are formed a structure ${}_nD_v$. For each monomial M we call I_M the position of M according to the ordering. Conversely, with M_I we denote the I-th monomial of the ordering. Finally, for an I with $M_I = x_1^{i_1} x_2^{i_2} \cdots x_v^{i_v}$, we define $F_I = i_1!i_2! \cdots i_v!$. Then an addition, a scalar multiplication and a vector multiplication in ${}_nD_v$ can be defined as follows:

$$\begin{aligned}(a_1, a_2, \cdots, a_N) + (b_1, b_2, \cdots, b_N) &= (a_1 + b_1, a_2 + b_2, \cdots, a_N + b_N) \\ t \cdot (a_1, a_2, \cdots, a_N) &= (t \cdot a_1, t \cdot a_2, \cdots, t \cdot a_N) \\ (a_1, a_2, \cdots, a_N) \cdot (b_1, b_2, \cdots, b_N) &= (c_1, c_2, \cdots, c_N)\end{aligned} \tag{1}$$

where t is an arbitrary real number, and the coefficients c_I are defined as follows:

$$c_I = F_I \cdot \sum_{\substack{0 \le J,K \le N \\ M_J \cdot M_K = M_I}} \frac{a_J \cdot b_K}{F_J \cdot F_K} \qquad (I = 1, 2, \ldots, N) \tag{2}$$

Differential calculus ∂_j can also be defined in ${}_nD_v$, for example, the first order derivative is defined as:

$$\partial_j(a_1, a_2, \cdots, a_N) = (d_1, d_2, \cdots, d_N) \qquad (j = 1, 2, \ldots, v) \tag{3}$$

where d_I $(I = 1, 2, \ldots, N)$ is equal to a_J (J is the ordinal number of the monomial $M_I \cdot x_j$) while the order of M_I is less than n; otherwise, d_I is equal to 0. With the existence of ∂_j operation as a kind of arithmetic operations, ${}_nD_v$ becomes a differential algebra.

2.2 Important Functions in Differential Algebra

Standard functions, such as exponentials and logarithmic and trigonometric functions can be generalized to differential algebra. In fact, all functions can be generalized straightforwardly [8].

Noting that for any differential algebraic vector of the form $(0, q_2, \cdots, q_N) \in {}_nD_v$, that is, with a zero in the component belonging to the zeroth order monomial, we have the following property:

$$(0, q_2, \cdots, q_N)^m = (0, 0, \ldots, 0) \qquad \text{for } m > n \tag{4}$$

which follows directly from the definition of the multiplication in ${}_nD_v$ defined in eq. (1).

Let us begin our discussion of special functions with the exponential function $\exp(x)$. Assume we have to compute the exponential of a differential algebraic vector that has already been created by previous operations. We note that the functional equation $\exp(x + y) = \exp(x) \cdot \exp(y)$ also holds in nonstandard analysis. As we will see, this facilitates the computation of the exponential. We obtain:

$$\begin{aligned} \exp[(a_1, a_2, \cdots, a_N)] &= \exp(a_1) \cdot \exp[(0, a_2, \cdots, a_N)] \\ &= \exp(a_1) \cdot \sum_{i=0}^{\infty} \frac{(0, a_2, \cdots, a_N)^i}{i!} \\ &= \exp(a_1) \cdot \sum_{i=0}^{n} \frac{(0, a_2, \cdots, a_N)^i}{i!} \end{aligned} \tag{5}$$

In the last step eq. (4) is used. This entails that the sum has to be taken through only order n, which allows the exponential computation in a finite number of steps.

A logarithm of a differential algebraic vector exists if and only if $a_1 > 0$. In this case one obtains:

$$\begin{aligned} \log[(a_1, a_2, \cdots, a_N)] &= \log\{a_1[1 + (0, \frac{a_2}{a_1}, \frac{a_3}{a_1}, \cdots, \frac{a_N}{a_1})\} \\ &= \log(a_1) + \sum_{i=1}^{\infty} (-1)^{i+1} \frac{1}{i} (0, \frac{a_2}{a_1}, \frac{a_3}{a_1}, \cdots, \frac{a_N}{a_1})^i \\ &= \log(a_1) + \sum_{i=1}^{n} (-1)^{i+1} \frac{1}{i} (0, \frac{a_2}{a_1}, \frac{a_3}{a_1}, \cdots, \frac{a_N}{a_1})^i \end{aligned} \tag{6}$$

Other fundamental functions, such as root function, sine and cosine, can be extended into ${}_nD_v$ by series expansion to a finite order. In general, suppose a function f has an addition theorem of the form:

$$f(a + b) = g_a(b) \tag{7}$$

and $g_a(b)$ can be written in a power series, then by the same reasoning its differential algebraic extension can be computed exactly in a finite number of steps.

3 Differential Algebraic Aberration Theory for Electron Optical Systems

The focusing and imaging properties of an electron optical system can be described by a transfer map [8]:

$$\boldsymbol{r}_f = \mathfrak{R}(\boldsymbol{r}_i, \boldsymbol{\delta}) \tag{8}$$

where $\boldsymbol{r}_f$ denotes the final coordinates of a particle to its initial coordinates $\boldsymbol{r}_i$, and $\boldsymbol{\delta}$ denotes the systemic parameters. The gradient of the map with respect to coordinates $\partial\mathfrak{R}/\partial\boldsymbol{r}$ is corresponding to aberrations, and $\partial\mathfrak{R}/\partial\boldsymbol{\delta}$ is corresponding to sensitivities. Except for the most trivial cases, it is impossible to find a closed analytic solution for the map $\mathfrak{R}$. It is usually expanded the map in a Tailor series around a reference trajectory, where the linear term denotes the Gaussian optical properties, the three cubed term denotes the third order aberration, and so on. The higher the order to which the terms of this Tailor series are taken, the more accurate the map is solved, but the complexity of the solving calculation increases dramatically as well. Therefore, this procedure is limited to lower medium orders. However, differential algebraic method presents a straightforward way to compute nonlinearity to arbitrary orders. Here no analytic formulas for derivatives must be derived, and the method is always accurate to machine precision independent of the order of the derivative, which is in sharp contrast to the methods of numerical differentiation. It will be a good idea to introduce differential algebraic method to the field of aberration analysis.

In an electron optical system, the transfer map $\mathfrak{R}$ can be expressed by electron trajectory equations. In a laboratory coordinate system (x, y, z), the electron trajectory equation in an electromagnetic field can be expressed as follows:

$$\begin{cases} x'' = & \frac{1}{2u}(1+x'^2+y'^2)(\frac{\partial u}{\partial x} - x'\frac{\partial u}{\partial z}) \\ & +\sqrt{\frac{-e}{2m_0u}}\sqrt{1+x'^2+y'^2}[x'(B_yx' - B_xy') - y'B_z + B_y] \\ y'' = & \frac{1}{2u}(1+x'^2+y'^2)(\frac{\partial u}{\partial y} - y'\frac{\partial u}{\partial z}) \\ & +\sqrt{\frac{-e}{2m_0u}}\sqrt{1+x'^2+y'^2}[y'(B_yx' - B_xy') + x'B_z - B_x] \end{cases} \tag{9}$$

During the differential algebraic operation, set the coordinate x, y, the slope x', y', and the field components u, B_x, B_y, B_z to be differential algebraic vectors, we can solve the eq. (9) by using the present numerical integrating methods such as the fourth order Runge-Kutta method. The results from the differential algebraic method take the form:

$$_n\begin{bmatrix} x_f \\ y_f \\ x'_f \\ y'_f \end{bmatrix} = \sum_{i,j,k,l=0\sim n}^{i+j+k+l=n} x_0^i y_0^j {x'}_0^k {y'}_0^l \begin{bmatrix} A_{ijkl} \\ B_{ijkl} \\ C_{ijkl} \\ D_{ijkl} \end{bmatrix} \tag{10}$$

where the prefix n indicates the calculation is up to n-order and the suffix f indicates the observation plane locates at $z = z_f$ with reference to the object

plane at $z = z_0$. This denotes Gaussian properties expression (while $n = 1$) and arbitrary order aberration expressions (while $n > 1$, n is an integer) by differential algebraic method.

4 Application: *Schiske*'s Model Electrostatic Electron Lens

Here we introduce an example of electrostatic electron lens to show the advantages of differential algebra used in high order aberration analysis. *Schiske*'s model is a widely studied model of electrostatic electron lenses, which axial electric field distribution is described by an analytic expression:

$$\phi(z) = \phi_0(1 - \frac{k^2}{1 + (z/a)^2}) \tag{11}$$

Table 1. The comparison of the Gaussian optical properties and C_s between differential algebraic results and the analytic solutions for *Schiske*'s model electrostatic lens

	M	M_s	$-1/f_i$	C_s
Analytic solutions	−1.633299180136	−0.612018921815	−3.217556194266	−1198.665836584
Results of differential algebraic method	−1.633299180119	−0.612018921804	−3.217556194246	−1198.665836548
Relative errors	1.04084×10^{-11}	1.79733×10^{-11}	6.21590×10^{-12}	3.00334×10^{-11}

($\phi_0 = 5\text{V}, k^2 = 0.5, a = 0.025\text{m}, z_0 = -0.5\text{m}$)

Table 2. Results of the third order geometric aberration coefficients for *Schiske*'s model electrostatic lens

(a) The third order geometric aberration coefficients in x-direction

Spherical aberration coefficients	A_{0030}	A_{0021}	A_{0012}	A_{0003}		
	−1198.665836548	0.0	−1198.665836548	0.0		
Coma coefficients	A_{1020}	A_{1011}	A_{1002}	A_{0120}	A_{0111}	A_{0102}
	−7174.059869613	0.0	−2391.353289871	0.0	−4782.706579742	0.0
Field curvature and astigmatism coefficients	A_{2010}	A_{1110}	A_{0210}	A_{2001}	A_{1101}	A_{0201}
	−14323.92239355	0.0	−4779.181961961	0.0	−9544.740431594	0.0
Distortion coefficients	A_{3000}	A_{2100}	A_{1200}	A_{0300}		
	−9540.879159025	0.0	−9540.879159025	0.0		

(b) The third order geometric aberration coefficients in y-direction

Spherical aberration coefficients	B_{0030}	B_{0021}	B_{0012}	B_{0003}		
	0.0	−1198.665836548	0.0	−1198.665836548		
Coma coefficients	B_{1020}	B_{1011}	B_{1002}	B_{0120}	B_{0111}	B_{0102}
	0.0	−4782.706579742	0.0	−2391.353289871	0.0	−7174.059869613
Field curvature and astigmatism coefficients	B_{2010}	B_{1110}	B_{0210}	B_{2001}	B_{1101}	B_{0201}
	0.0	−9544.740431594	0.0	−4779.181961961	0.0	−14323.92239355
Distortion coefficients	B_{3000}	B_{2100}	B_{1200}	B_{0300}		
	0.0	−9540.879159025	0.0	−9540.879159025		

($\phi_0 = 5\text{V}, k^2 = 0.5, a = 0.025\text{m}, z_0 = -0.5\text{m}$)

Using a rotational coordinate system, the Gaussian properties are described by a first order transfer map [9]:

$$\begin{bmatrix} x_g \\ y_g \\ x'_g \\ y'_g \end{bmatrix} = \begin{bmatrix} M & & & \\ & M & & \\ -\frac{1}{f_i} & & M_s & \\ & -\frac{1}{f_i} & & M_s \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ x'_0 \\ y'_0 \end{bmatrix} \tag{12}$$

where (x_g, y_g, x'_g, y'_g) is the vector containing positions and slope on the Gaussian image plane, (x_0, y_0, x'_0, y'_g) is the vector containing positions and slope on the object plane.

Now we use differential algebraic method to calculate Gaussian properties and third order aberrations of *Schiske*'s model electrostatic lens. The variables x, y, x', y' are set to be differential algebraic vectors. We can solve the trajectory equations (9) by performing Runge-Kutta method and gain the differential algebraic vectors x_g, y_g, x'_g, y'_g in the Gaussian imaging plane. Therefore, the Gaussian optical properties and arbitrary high order aberrations can be obtained by the differential algebraic method shown in eq. (10).

We calculate a real *Schiske*'s model electrostatic lens with the parameters: $\phi_0 = 5\text{V}$, $k^2 = 0.5$, $a = 0.025\text{m}$, the object plane locates at $z_0 = -0.5\text{m}$. The comparison of the Gaussian optical properties and the third order spherical aberration coefficient C_s between differential algebraic results and the analytic solutions are shown in Table 1. From the relative errors of the two methods, it is proved that the differential algebraic method has very high accuracy.

All the coefficients of the third order geometric aberrations are calculated by differential algebraic method shown in Table 2.

5 Conclusion

In this paper, differential algebraic aberration method for electron optical systems is presented. By employing the effective tool, the arbitrary high order aberrations can be calculated with extreme high accuracy up to the machine precision. As an example, an important analytical model of electrostatic lenses named *Schiske*'s model lens has been studied, and the Gaussian properties and third order geometric aberration coefficients have been calculated. The results show that differential algebraic method is an effective tool with excellent accuracy for the aberration analysis and calculation of electrostatic electron lenses. This developed method can be of great utility in high order aberration analysis and computation for charged particle optical systems.

References

1. Xie, X., Liu, C. L.: Any order approximate analytical solutions of accelerator nonlinear dynamic system equations. Chinese Journal of Nuclear Science and Engineering **10** (1990) 273–276

2. Ximen, J. Y.: Canonical aberration theory in electron optics. J. Appl. Phys. **68** (1990) 5963–5967
3. Dragt, J., Forest, E.: Lie algebra theory of charged-particle optics and electron microscopes. Adv. in Electronics and Electron Phys. **67** (1986) 65–120
4. Hawkes, P. W., Kasper, E.: Principles of Electron optics, Volume 2. Academic Press, London 1989
5. Robinson, A., in: Proceedings of the Royal Academy of Sciences Ser A64. Amsterdam: North-Holland, **B64** (1961) 432–440
6. Niven, I., Formal power series, American Mathematical Monthly, **76–8** (1969) 871
7. Rall, L. B., The arithmetic of differentiation, Mathematics Magazine **59** (1986) 275–282
8. Berz, M., Differential algebraic description of beam dynamics to very high orders. Particle Accelerators **24** (1989) 109–124
9. Hawkes, P. W., Kasper, E.: Principles of Electron optics, Volume 1. Academic Press, London 1989

Optimizing Symmetric FFTs with Prime Edge-Length

Edusmildo Orozco[1] and Dorothy Bollman[2]

[1] Doctoral Program in CISE, UPRM
Mayagüez, Puerto Rico
eorozco@cs.uprm.edu
[2] Department of Mathematics, UPRM
Mayagüez, Puerto Rico
bollman@cs.uprm.edu

Abstract. It is known that a multidimensional FFT with prime edge-length p and linear symmetries in its inputs, given by a matrix S, can be computed efficiently in terms of cyclic convolutions by determining a nonsingular matrix M that commutes with S and that minimizes the number of $MS-$orbits. To date the only known method for determining such an M is by exhaustion, which takes time $O(p^6)$ in the two-dimensional case and time $O(p^{12})$ in the three-dimensional case. In this work we study methods for determining M directly. Our results include algorithms which, assuming the availability of primitive polynomials, compute M in time $O(p)$ in the two-dimensional and, in a special three-dimensional case that is important for crystallographers. Furthermore, also assuming the availability of primitive polynomials of degree three, we give an $O(p^3)$ time algorithm to compute the $M-$minimal three-dimensional case.

Keywords: Symmetric FFT, cyclic convolution, finite field, orbit.

1 Introduction

For some data intensive problems, for instance, x-ray crystal diffraction intensity analysis, reductions in the amount of data can make a significant difference even though the arithmetic complexity remains the same. These reductions are induced by structured redundancy patterns in the input, which in turn induce redundancies in the output. Such a problem, which has recently received attention [7], [8], [9], is the problem of making more efficient the computation of multidimensional fast Fourier transforms with linear symmetries. In the rest of this section we outline this problem.

For the purposes of this paper it suffices to think of the d-dimensional discrete Fourier transform (DFT) with edge-length N as simply a function

$$f : \mathcal{A}_{d,N}(C) \to \mathcal{A}_{d,N}(C)$$

where C denotes the set of complex numbers and $\mathcal{A}_{d,N}(C)$ denotes the set of d-dimensional arrays with edge-length N over C. The time required to compute

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 736–744, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the DFT with edge length N using the definition is $O(N^{2d})$. However, the *fast* Fourier transform ("FFT") can be computed in time $O(N^d log\ N)$. The input of a DFT or FFT is a complex-valued mapping f defined on $Z_N^d = Z_N \times \cdots \times Z_N$, d times, where Z_N denotes the integers modulo N. A *linear* symmetry on such a function f is defined as a $d \times d$ nonsingular matrix S over Z_N such that $f(\mathbf{k}) = f(S\mathbf{k})$ for all $\mathbf{k} \in Z_N^d$. Of particular interest are the linear symmetries in three-dimensional crystallographic FFTs.

Let us consider a two-dimensional example. The mapping f defined on Z_5^2 by the matrix

$$f = \begin{pmatrix} 2.9 & 2.3 & 1.5 & 1.5 & 2.3 \\ 1.2 & 6.0 & 4.3 & 4.6 & 2.8 \\ 1.4 & 3.3 & 5.1 & 4.2 & 1.7 \\ 1.4 & 1.7 & 4.2 & 5.1 & 3.3 \\ 1.2 & 2.8 & 4.6 & 4.3 & 6.0 \end{pmatrix} \tag{1}$$

is S-symmetric where

$$S = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} \tag{2}$$

(We assume that rows and columns are numbered 0, 1, 2, 3, 4.) For instance, if we let $\mathbf{k} = (2, 1)$, then $S\mathbf{k} = (-2, -1) = (3, 4)$. Thus, $f(\mathbf{k}) = f(S\mathbf{k}) = 3.3$. Linear symmetries S in the inputs induce linear symmetries S_* in the outputs, where S_* denotes the transpose of the inverse of S. The relation $\approx_S$ on Z_N^d defined by $\mathbf{a} \approx_S \mathbf{b}$ if and only if $S^i\mathbf{a} = \mathbf{b}$ for some integer i is an equivalence relation and the equivalence class $O_S(\mathbf{a})$ containing $\mathbf{a} \in Z_N^d$ is called an $S-$*orbit*. A set of representatives of the $S-$orbits is called a *fundamental set* $\mathcal{F}_\mathcal{S}$.

For example, a fundamental set for the $S-$orbits induced by the symmetry matrix S over Z_5 given above is

$$\{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(1,3),(1,4),(2,0),(2,1),(2,2),(2,3),(2,4)\}.$$

An $S-$symmetric function f is constant on each $S-$orbit and is thus completely determined by its values on a fundamental set. Auslander and Shenefelt [1] have shown that when $N = p$ is prime, a fundamental set can be reordered by a generator g of the multiplicative cyclic group of Z_p in such a way that the FFT can be computed in terms of cyclic convolutions. Efficiency increases with decreasing number of cyclic convolutions. In [7] it is shown that the number of cyclic convolutions can be decreased if, instead of reordering fundamental sets by a generator of the cyclic group of Z_p, we reorder via a $d \times d$ nonsingular matrix M that commutes with S. Such a matrix M induces an equivalence relation $\approx_M$ on the set of $S-$orbits, or equivalently on $\mathcal{F}_\mathcal{S}$, defined by $O_S(\mathbf{a}) \approx_M O_S(\mathbf{b})$ if and only if $M^i\mathbf{a} = M^j\mathbf{b}$ for some integers i and j. We call the equivalence classes induced by $\approx_M$ $MS-$*orbits*.

The problem of minimizing the number of cyclic convolutions now becomes the problem of, given S, choosing an M commuting with S that produces the minimal number of $MS-$orbits. To date, the only known method for choosing M is by exhaustion. However, this is very costly. The time for computing M

by exhaustion is $O(p^6)$ in the two dimensional case and is $O(p^{12})$ in the three dimensional case. In this paper we study methods for computing M directly.

The organization of the rest of the paper is as follows. In Section 2, we present mathematical preliminaries that are needed in the rest of the work. In Section 3, we characterize those symmetry matrices S for which an M can be obtained that gives exactly one $MS-$orbit, we indicate how such an M can be calculated, and we give specific algorithms for the two-dimensional and three-dimensional cases. In Section 4, we give a complete solution to the $MS-$problem for the two-dimensional case. In Section 5, we give solution to a special three-dimensional case which is important for crystallographers.

2 Mathematical Preliminaries

From now on, p will denote a prime, I_n the $n \times n$ identity matrix, and $\phi_A(x)$ the characteristic polynomial $det(xI - A)$ of a square matrix A.

For any nonsingular $n \times n$ square matrix A there exists a positive integer m such that $A^m = I_n$. The smallest such m is called the *period* of A.

The *order* of an element a of a finite field $GF(p^n)$ is defined to be the least positive integer m such that $a^m = 1$. If $g \in GF(p^n)$ is a generator of the multiplicative group of $GF(p^n)$, the *index* of a with respect to g is the smallest positive integer $ind_g(a)$ for which $g^{ind_g(a)} = a$. A *primitive polynomial* of degree n over Z_p is an irreducible polynomial $P(x)$ that has a root that generates the multiplicative group of $GF(p^n)$.

There are various representations of a finite field $GF(p^n)$. In our work it is useful to consider the following three:

1. $K_1(p^n) = \{a_{n-1}\alpha^{n-1} + a_{n-2}^{n-2} + \cdots + a_1 + a_0 | a_i \in Z_p\}$, where α is a root of an irreducible polynomial of degree n over Z_p.
2. $K_2(p^n) = \{0\} \cup \{\alpha^i | i = 0, 1, \cdots, p^n - 1\}$ where α is a root of a primitive polynomial of degree n over Z_p.
3. $K_3(p^n) = \{a_{n-1}S^{n-1} + a_{n-2}S^{n-2} + \cdots + a_1 S + a_0 I_n | a_i \in Z_p\}$ where S is an $n \times n$ matrix over Z_p with irreducible characteristic polynomial.

These three representations are isomorphic to each other and it is useful to examine the mappings that give these isomorphisms.

Theorem 1. Let $P(x)$ be a primitive polynomial over Z_p and let β be root of $P(x)$ in $GF(p^n)$. Also, let $\alpha \in GF(p^n)$ be a root of an irreducible polynomial $R(x)$ of degree n over Z_p. Then there exists $Q(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ over Z_p such that $P(Q(\alpha)) = 0$. Let S be a nonsingular $n \times n$ over Z_p. Define $h_1(\beta^i) = Q(\alpha)^i$ and $h_2(\beta^i) = Q(S^i)$ for each $i = 0, 1, \cdots, p^n - 1$. Then $h_1 : K_2(p^n) \to K_1(p^n)$ and $h_2 : K_2(p^n) \to K_3(p^n)$ are isomorphisms.

Corollary. Let S be a nonsingular $n \times n$ matrix over Z_p with irreducible characteristic polynomial $\phi_S(x)$. Then $Q(S)$ commutes with S and has maximal period $p^n - 1$.

Let $P(c_0 + c_1x + \cdots + c_{n-1}x^{n-1})\ mod\ R(x) = e_0 + e_1x + \cdots + e_{n-1}x^{n-1}$. Then each e_i is a polynomial in $c_0, c_1, \cdots, c_{n-1}$ and the solution $c_0, c_1, \cdots, c_{n-1}$ of the systems of polynomial congruences $e_i = 0\ mod\ p$ gives the matrix $M = Q(S) = c_0 + c_1S + \cdots + c_{n-1}S^{n-1}$ of period $p^n - 1$ that commutes with S.

3 The M-Minimal Case

The ideal symmetry is one for which the DFT can be computed via just one cyclic convolution. We call an $n \times n$ matrix S over Z_p $M-$*minimal* if there exists an $n \times n$ matrix M over Z_p for which there is exactly one nontrivial $MS-$orbit. Given S, we say that M is *optimal* for S if the number of nontrivial $MS-$orbits induced by M is minimal. $M-$minimal matrices are characterized by the following

Theorem 2. An $n \times n$ matrix S is $M-$minimal if and only if S is similar to $\begin{pmatrix} A_m & \cdots & 0 \\ \vdots & \ddots & \\ 0 & & A_m \end{pmatrix}$, where $\phi_{A_m}(x)$ is an $m-$degree irreducible polynomial over Z_p.

In particular, any scalar matrix S, i.e., $S = aI_n$ for some $a \in Z_p$, and any S with $\phi_S(x)$ irreducible are $M-$minimal. Theorems 3 and 4 show how to find an optimal M in each of these two cases.

Theorem 3. For any $a \in Z_p$, $S = aI_n$ is $M-$minimal and the companion matrix M of any primitive polynomial is optimal for S.

Example 1

$$S = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{3}$$

is $M-$minimal over Z_p for any p. If, for example, $p = 47$, then

$$M = \begin{pmatrix} 0 & 0 & 43 \\ 1 & 0 & 46 \\ 0 & 1 & 0 \end{pmatrix} \tag{4}$$

is optimal for S since $x^3 + x + 4 = x^3 - 46x - 43\ mod\ 47$ is primitive over Z_{47}.

Theorem 4. Every $n \times n$ matrix S over Z_p with irreducible characteristic polynomial is $M-$minimal and the matrix $M = Q(S)$ defined in the corollary to Theorem 1 is optimal for S.

Theorem 5. An $n \times n$ matrix S over Z_p where $n = 2$ or $n = 3$ is $M-$minimal if and only if S is scalar or $\phi_S(x)$ is irreducible.

In the $2-$dimensional case, it is easy to describe the solution of the system of congruences that gives the coefficients of the polynomial $Q(x)$.

Theorem 6. Let $P(x) = x^2 + ax + b$ be any primitive polynomial over Z_p and let S be a 2×2 matrix over Z_p with $\phi_S(x) = x^2 + cx + d$ irreducible. Then $\frac{a^2-4b}{c^2-4d}$ is a quadratic residue e mod p and $M = eS + fI_2$, where $f = 2^{-1}(ec - a)$, is optimal for S.

Example 2. Let S be the matrix over Z_{97} defined by

$$S = \begin{pmatrix} 23 & 65 \\ 84 & 10 \end{pmatrix} \tag{5}$$

Then $\phi_S(x) = x^2 + 64x + 8$. The polynomial $P(x) = x^2 - x + 5$ is primitive over Z_{97}. Using the equations in Theorem 6, we find that $e^2 = 31 \pmod{97}$ and so $e = 15$ and $f = 44 \pmod{97}$. An optimal matrix for S is thus,

$$M = 15S + 44I_2 = \begin{pmatrix} 1 & 5 \\ 96 & 0 \end{pmatrix} \tag{6}$$

In the $n-$dimensional case where $n \geq 3$, the determination of the coefficients of $Q(x)$ is more elusive. When $n = 3$, we can show that for a 3×3 symmetriy matrix S with irreducible characteristic polynomial $\phi_S(x) = x^3 + dx^2 + ex + f$, then a matrix M that is $M-$minimal for S is $gS^2 + hS + iI_3$ where g, h, i is the solution of the system of congruences

$$c+adfg^2+d^3fg^3-2defg^3+f^2g^3-2afgh-3d^2fg^2h+3efg^2h+3dfgh^2-fh^3+bi+3dfg^2i \\ -6fghi + ai^2 + i^3 = 0 \ mod\ p$$

$$adeg^2 - afg^2 + d^3eg^3 - 2de^2g^3 - d^2fg^3 + 2efg^3 + bh - 2aegh - 3d^2eg^2h + 3e^2g^2h \\ +3dfg^2h + 3defgh^2 - 3fgh^2 - eh^3 + 3deg^2i - 3fg^2i + 2ahi - 6eghi + 3hi^2 = 0 \ mod\ p$$

$$bg + ad^2g^2 - aeg^2 + d^4g^3 - 3d^2eg^3 + e^2g^3 + 2dfg^3 - 2adgh - 3d^3g^2h + 6deg^2h - 3fg^2h \\ ah^2 + 3d^2gh^2 - 3egh^2 - dh^3 + 2agi + 3d^2g^2i - 3eg^2i - 6dghi + 3h^2i + 3gi^2 = 0 \ mod\ p$$

We know of no method to solve this system of polynomial congruences. However, because of the isomorphism between $K_1(p^3)$ and $K_2(p^3)$ we are guaranteed that a solution exists. Indeed, since g, h, i is a solution if and only if $P(g\alpha^2 + h\alpha + i) = 0$, where $\phi_S(\alpha) = 0$, and P is of degree 3, there are exactly three solutions. One idea is to simply use trial and error to determine one such solution g, h, i. We have written a program in C for this purpose. The time required for this method is $O(p^3)$.

Example 3. Let $S = \begin{pmatrix} 86 & 36 & 87 \\ 43 & 8 & 90 \\ 78 & 43 & 8 \end{pmatrix}$

be a symmetry matrix over Z_{97}. Then $\phi_S(x) = x^3 + 92x^2 + 3x + 96$ is irreducible. Using our program with the primitive polynomial $P(x) = x^3 + x + 7$, we find three solutions to the above system of conguences: $\{g, h, i\} = \{7, 62, 14\}$,

$\{g, h, i\} = \{11, 13, 38\}$, and $\{g, h, i\} = \{79, 22, 45\}$. Using the first solution, we find that a matrix optimal for S is $M = 7S^2 + 62S + 14I_3 = \begin{pmatrix} 26 & 18 & 57 \\ 75 & 84 & 40 \\ 39 & 75 & 84 \end{pmatrix}$.

4 Two Dimensions

The question remains of how to choose an optimal M for a symmetry matrix S that is not necessarily $M-$minimal. In this section we completely solve the problem for two dimensions. We characterize the various cases according to the factorability of $\phi_S(x)$.

Theorems 3 and 5 give us the results for $S = \lambda I_2$ and $\phi_S(x)$ irreducible, respectively. The following two theorems cover the remaining cases in two dimensions.

Theorem 7. If $\phi_S(x) = (x - \lambda)^2$ but $S \neq \lambda I_2$, then an optimal M for S is gI_2.

Now, let S be a nonsingular matrix such that $\phi_S(x) = (x - \lambda_1)(x - \lambda_2)$, $\lambda_1 \neq \lambda_2$. Also, let $e_i = \dfrac{\text{lcm}(\text{ind}_g(\lambda_1), \text{ind}_g(\lambda_2))}{\text{ind}_g(\lambda_i)}$, $i = 1, 2$, k_i be the order of λ_i and $l = \gcd(k_1, k_2)$. Define $R(\lambda_1, \lambda_2) = \{t \,|\, \gcd(e_1 - e_2 t, l) = 1\}$. It can be shown that such an $R(\lambda_1, \lambda_2)$ is nonempty. Finally, let

$$\beta(\lambda_1, \lambda_2) = g^{t_0},$$

where $\gcd(\text{ind}_g(\lambda_1), t_0)$ is the minimum value of

$$\{\gcd(\text{ind}_g(\lambda_1), t) \mid t \in R(\lambda_1, \lambda_2)\}.$$

(that is, $\text{ind}_g(\beta(\lambda_1, \lambda_2)) = t_0$.)

Theorem 8. Let $\phi_S(x) = (x - \lambda_1)(x - \lambda_2)$, where $\lambda_1 \neq \lambda_2$, k_1 and k_2 are the orders of λ_1 and λ_2, respectively. Then, assuming $k_1 \leq k_2$, an optimal matrix M for S is $M = a'S + b'I_2$, where $a' = \frac{g - \beta(\lambda_1, \lambda_2)}{\lambda_1 - \lambda_2}$, $b' = g - a'\lambda_1$, and g is a generator of the cyclic group of Z_p.

We summarize the results for choosing an optimal M in the 2−dimensional case in the following

Algorithm

1. **Inputs:** $S = \begin{pmatrix} s_{00} & s_{01} \\ s_{10} & s_{11} \end{pmatrix}$ and a prime p
   Output: optimal matrix M
2. Compute $\phi_S(x) = x^2 + cx + d$, where $c = -(s_{00} + s_{11}) \pmod p$ and $d = s_{00}s_{11} - s_{01}s_{10} \pmod p$.
3. Find the roots of $\phi_S(x)$
   3.1 If $\phi_S(x)$ is irreducible, find e such that $e^2 = \frac{a^2 - 4b}{c^2 - 4d} \pmod p$ and let $f = 2^{-1}(ec - a) \pmod p$. Set $M = eS + fI_2$.

3.2 If $\phi_S(x) = (x-\lambda)^2$

3.2.1 If $S - \lambda I_2 = 0$, then $M = \begin{pmatrix} 0 & -a \\ 1 & -b \end{pmatrix}$, where $P(x) = x^2 + ax + b$ is a primitive polynomial over Z_p.

3.2.2 If $S - \lambda I_2 \neq 0$, then $M = gI_2$, where $g = P(0) = b$ is a generator of Z_p.

3.3 If $\phi_S(x) = (x-\lambda_1)(x-\lambda_2)$, $\lambda_1 \neq \lambda_2$, then $M = a'S + b'I_2$, where a' and b' are computed according to Theorem 8.

In order to implement this algorithm, we make use of a precomputed table of quadratic primitive polynomials. It is well known [5] that for any $n-$degree primitive polynomial $P(x)$ over Z_p, the constant $(-1)^n P(0)$ is a generator g of the multiplicative cyclic group of Z_p. Thus, having a precomputed table of primitives also gives us generators for the cyclic group of Z_p. Now it is easy to show that, assuming the availability of primitive polynomials, each step of the algorithm takes either constant or $O(p)$ time.

The characteristic polynomial $\phi_S(x)$ can be computed in constant time and its roots can be determined in time $O(p)$. The calculation of M in Steps 3.1 and 3.2 requires constant time. The primitive polynomial in Step 3.2.1 can be found by table lookup and thus time $O(p)$ (or time $O(log\ p)$ using binary search for tables with a very large number of primitive polynomials). The calculation of M in Step 3.3 requires time $O(p)$.

Example 4. Let the symmetry matrix S be defined over Z_{379} by

$$S = \begin{pmatrix} 82 & 77 \\ 296 & 316 \end{pmatrix}$$

The characteristic polynomial of S is $\phi_S(x) = (x-11)(x-8)$. A primitive element for $Z_{379} - \{0\}$ is $g = 2$ and, hence $\text{ind}_2(11) = 217$ and $\text{ind}_2(8) = 3$. The orders of $\lambda_1 = 11$ and $\lambda_2 = 8$ are $k_1 = 54$ and $k_2 = 126$, respectively. Also, $e_1 = 3$, $e_2 = 217$, and $l = 18$. In this case, M is optimal if $\gcd(3 - 217\ \text{ind}_2(\beta), 18) = 1$ and $\gcd(217,\ \text{ind}_2(\beta)) = 1$. Let $\text{ind}_2(\beta) = 2$. Thus $\beta = 4$ and $a' = (2-4) * (11-8)^{-1} = 252$ and $b' = 2 - 252 * 11 = 262$. An optimal M for S is given by

$$M = 252S + 262I_2 = \begin{pmatrix} 81 & 75 \\ 308 & 304 \end{pmatrix}.$$

5 A Three-Dimensional Case

In this section we give a solution to a special three-dimensional case which is important for crystallographers [1]. The following theorem outlines the procedure to compute the optimal M for such an S.

Theorem 9. Let S be such that $\phi_S(x) = (x^2 + cx + d)(x - \lambda)$ where $\phi_{S'}(x) = x^2 + cx + d$ is irreducible and $\lambda \neq 0 \in Z_p$. Then, an optimal matrix M for S

is $M = c_2S^2 + c_1S + c_0I_3$, where $c_2 = (\phi_{S'}(\lambda))^{-1}(g - e\lambda - f)$, $c_1 = e + cc_2$, $c_0 = f + dc_2$, g is a generator of Z_p and e and f are as in theorem 6.

Example 5. Let us consider the symmetry $S = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ over Z_N, which generates the point group P_3 (according to the notation used by crystallographers.) The characteristic polynomial of S is $\phi_S(x) = (x-1)(x^2+x+1)$. Matrix S is similar to $\begin{pmatrix} S' & 0 \\ 0 & 1 \end{pmatrix}$, where $S' = \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}$ is the companion matrix associated to the polynomial $\phi_{S'}(x) = x^2 + x + 1$.

Let us assume that $N = p$ is a prime. Table 1 shows the values of $p \leq 359$ for which $\phi_{S'}(x)$ happens to be irreducible.

Table 1. Values of primes $p \leq 359$ for which $\phi_{S'}(x) = x^2 + x + 1$ is irreducible

2	5	11	17	23	29	41	47	53	59	71	89	107	113	131	137	149	167
173	179	191	197	227	233	239	251	257	263	269	281	293	311	317	347	353	359

For instance, if we set $p = 5$, then, $c = 1$, $d = 1$ and $\phi_{S'}(\lambda) = \phi_{S'}(1) = 3$. Now, applying Theorem 6, we compute $e = 2$ and $f = 3$. Hence, $c_2 = (3)^{-1}(2 - 2 * 1 - 3) = 4$, $c_1 = 2 + 1 * 4 = 1$, and $c_0 = 3 + 1 * 4 = 2$. Therefore, an optimal matrix M for S is $M = 4S^2 + S + 2I_3 = \begin{pmatrix} 1 & 2 & 0 \\ 3 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$.

6 Conclusions and Future Work

The computation via cyclic covolutions of a multidimensional FFT with linear symmetries can be optimized by minimizing the number of $MS-$orbits. The ideal case is when S is $M-$minimal, i.e., there exists an M for which there is exactly one $MS-$orbit. We have given necessary and sufficient conditions for the symmetry matrix S to be $M-$minimal. In particular, S is $M-$minimal if S is scalar or has irreducible characteristic polynomial. In the two- and three-dimensional cases, these are the only $M-$minimal symmetry matrices. We show how to compute an optimal M for scalar S in general and for $\phi_S(x)$ irreducible when the number n of dimensions is 2 or 3. Assuming the availability of primitive polynomials, M can be computed in time $O(p)$ when $n = 2$ and in time $O(p^3)$ when $n = 3$. For $n = 2$ we give an algorithm which, making use of a precomputed table of primitive polynomials, computes for any given symmetry matrix S an optimal M in time cp. For $n = 3$, an important special case is when $\phi_S(x) = (x-\lambda)(x^2+cx+d)$. For this, we give a procedure, based upon the two-dimensional irreducible case, to find the optimal M for S.

We are presently working on a general algorithm for determining M in the three-dimensional case for arbitrarily given S, not just the two cases mentioned

above. We would also like to determine a general solution g, h, i for the system of congruences in Section 3 in terms of the prime p and the coefficients of $\phi_S(x)$ and a cubic primitive polynomial over Z_p, thus replacing the $O(p^3)$ search by a constant time calculation.

Acknowledgements. The work of the first author was supported by the National Science Foundation under Grant No. 9817642.

References

1. Auslander, L., Shenefelt, M.: Fourier Transforms that Respect Crystallographic Symmetries. IBM J. Res. and Dev. 31 (1987) 213–223
2. Elspas, B.: The Theory of Autonomous Linear Sequential Networks, Linear Sequential Switching Circuits. (eds.): W. Kautz, Holden-Day Inc. (1965) 21–61
3. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press. (1999)
4. McCoy, N.H.: Rings and Ideals. The Carus Mathematical Monographs. The Mathematical Association of America (1956)
5. Lidl,R., Niederreiter, H.: Finite Fields. Encyclopedia of Mathematics and its Applications, Vol. 20. 2nd edn. Cambridge University Press. (1997)
6. Orozco, E., Bollman, D., Seguel, J., Moreno, O.: Organizing Crystallographic Data. Poster presentation. 1st Conference in Protein Structure, Function and Dynamics. Feb 7–9 (2003). Ponce, P.R.
7. Seguel, J., Bollman, D., Orozco, E.: A New Prime Edge-Length Crystallographic FFT. In: Sloot, P., Tan, C., Dongarra, J., Hoekstra, A. (eds.): Lecture Notes in Computer Science, Springer-Verlag, Part II. 2330 (2002) 548–557
8. Seguel, J.: Design and Implementation of a Parallel Prime Edge-Length Symmetric FFT. In: Kumar V. et al (eds.): Lecture Notes in Computer Science, Springer-Verlag, 2667 (2003) 1025–1034
9. Seguel J., Burbano, D.: A Scalable Crystallographic FFT. In: Dongarra, J., Laforenza, De., Orlando S.: (eds.): Euro PVM/MPI 2003, Lecture Notes in Computer Science, 2840 (2002) 134–141

A Spectral Technique to Solve the Chromatic Number Problem in Circulant Graphs

Monia Discepoli[1,2], Ivan Gerace[1], Riccardo Mariani[1], and Andrea Remigi[1]

[1] Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, via Vanvitelli 1, I-06123 PG, Italia.
[2] Dipartimento di Matematica "Ulisse Dini", Università degli Studi di Firenze, viale Morgagni 67a, I-50134 FI, Italia.

Abstract. The computation of the CHROMATIC NUMBER of circulant graph is essentially hard as in the general case. However in this case it is possible to use spectral properties of the graph to obtain a good coloration. In this paper we use these properties to construct two heuristic algorithms to color a circulant graph. In the case of sparse graphs, we show that our heuristic algorithms give results better than the classical ones.

Keywords: Circulant Graphs, Chromatic Number, Graph Coloring, Spectral Properties of Graphs, Approximation Algorithms.

1 Introduction

Circulant matrices are an important class of matrices. Indeed both the linear algebra and the combinatorics scientists have studied the properties of this class of matrices [6,7,8,9,11,12]. In particular algebraic properties of circulant matrices turn to be very useful to construct efficient algorithm in many applications. *Circulant graphs* are graphs whose adjacent matrix is circulant. They have several applications in areas like telecommunication networks, VLSI design and distributed computing [4,15,16]. This relevance to distributed computing is due to the fact that circulant graph is a natural extension of a ring, with increased connectivity.

The CHROMATIC NUMBER is the minimum number of colors by means of which it is possible to color a graph in such a way that each vertex has a different color with respect to the adjacent vertices. Such a problem is an NP-hard problem [14] and is even hard to obtain a good approximation of the solution in a polynomial time [17]. Although in a lot of computational problems the cost decreases when these problems are restricted to circulant graphs [6,9], the CHROMATIC NUMBER problem is NP-hard even restrecting to circulant graphs [9]. Moreover the problem of finding a good approximation of the CHROMATIC NUMBER problem on circulant graphs is also NP-hard.

In [9] it is shown how the signs of the eigenvectors of a circulant graph can be used to obtain a good coloration of the graph it-self. However there is still a problem to find the right eigenvectors to obtain the best coloration. In this

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 745–754, 2004.
© Springer-Verlag Berlin Heidelberg 2004

paper we propose two different heuristics to choose the best set of eigenvectors: by means of this, we find a correct coloration of the graph.

To estimate the performance of our algorithms we compare the results with the ones obtained by the Johnson algorithm, that is one of the most simple and efficient to solve the CHROMATIC NUMBER problem [3].

When the graph is sparse spectral techniques give closer bounds in the 75% of the cases.

2 Spectral Properties of Graphs

Let $G = \langle V, E \rangle$ be a non-oriented graph, where V is the finite set of its vertices and E is the set of its edges, which are pairs of vertices. In particular let $\{v_1, v_2, \ldots, v_n\}$ the vertices of graph G; if $\{v_i, v_j\}$ is an edge, then v_i and v_j are called *adjacent*.

The *adjacency matrix* of a graph is the matrix $A = A(G)$, whose entries are the following

$$a_{ij} = \begin{cases} 1, & \text{if } \{i,j\} \in E \\ 0, & \text{otherwise.} \end{cases}$$

In the case of non-oriented graph we have that the corresponding adjacency matrix is symmetric. The indices of rows and columns of the matrix A correspond to an arbitrary labelling of the vertices of G. For this reason we are interested in properties of the adjacency matrix that are invariant respect to permutations of rows and columns. The spectrum of the matrix A is invariant under permutations of matrix indices.

We denote as *spectrum of a graph* G the spectrum of the corresponding adjacency matrix.

Let us assume that the connected graph G has n vertices. Then its spectrum satisfies the following important properties [10]:

1. the eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{n-1}$ of A are real (labelled with $\lambda_0 > \lambda_1 \geq \ldots \geq \lambda_{n-1}$);
2. the corresponding eigenvectors $\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n-1}$ can be chosen to be orthonormal;
3. $\sum_{i=0}^{n-1} \lambda_i = 0$;
4. the maximum eigenvalue λ_0 is the spectral radius of A and is simple;
5. $\boldsymbol{u}_0$ can be chosen with positive components.

3 Circulant Graphs

A matrix $A \in \mathbb{R}^{n \times n}$ is said to be *circulant* if its entries satisfy $a_{i,j} = a_{0,j-i}$, where the indices are reduced modulo n and belong to the set $\{0, 1, \ldots, n-1\}$.

In other words, the $i-$th row of A is obtained from the first row of A by a cyclic shift of $i-1$ steps. So any circulant matrix is determined by its first row.

Let $\boldsymbol{a} = [a_0\, a_1 \ldots a_{n-1}]$ be the first row of A. The eigenvalues of A are

$$\lambda_j = \sum_{i=1}^{n-1} a_j \omega^{ji}, \qquad j = 0, 1, \ldots, n-1,$$

where $\omega = e^{\frac{2\pi\iota}{n}}$, $(\iota = \sqrt{-1})$, is a primitive n-th root of the unity. Moreover the eigenvectors of A are

$$\boldsymbol{u}_j = \begin{bmatrix} 1 & \omega^j & \omega^{2j} & \omega^{3j} & \ldots & \omega^{(n-1)j} \end{bmatrix}^T.$$

Note that every circulant matrix has the same eigenvectors.

A *circulant graph* is a graph with circulant adjacency matrix. The class of circulant graphs is a subset of the class of vertex -symmetric graphs and is exactly the class of the Cayley graphs of a finite cyclic group. A connected graph G with a prime number of vertices is vertex-symmetric if and only if it is a circulant graph [18].

Since the adjacency matrix of a graph is a symmetric matrix with zero entries on the diagonal, it follows that $a_0 = 0$ and $a_i = a_{-i}$ (i.e., $a_i = a_{n-i}$) $(1 \leq i \leq n-1)$. Note that if a circulant graph is not connected, it is composed of isomorphic circulant components.

3.1 Spectral Properties of Circulant Graphs

Let G be a circulant graph of degree d with n vertices and adjacency matrix A. In this case the eigenvalues of A are real and we can choose a real basis of eigenvectors as follows:

$$\lambda_0 = \sum_{k=1}^{n-1} a_k,$$

$$\lambda_j = \lambda_{n-j} = \sum_{k=1}^{n-1} a_k \cos\left(\frac{2jk\pi}{n}\right) \qquad 1 \leq j \leq n-1.$$

Note that though we do not assume that the eigenvalues are ordered, λ_0 is always the largest eigenvalue, that is the spectral radius. The eigenspace corresponding to the eigenvalues λ_j $(1 \leq j \leq n-1)$, except for $\lambda_{\frac{n}{2}}$, have dimension at least 2. That allows to find a basis of real eigenvectors.

For the rest of the paper we use the following real eigenvectors. The eigenvector related to $\lambda_0 = d$ is

$$\boldsymbol{u}_0 = [1 \quad 1 \quad 1 \quad \ldots \quad 1]^T,$$

and the eigenvectors related to λ_j and λ_{n-j} will be given by

$$\boldsymbol{u}_j = \left[1 \quad \cos\left(j\frac{2\pi}{n}\right) \quad \cos\left(2j\frac{2\pi}{n}\right) \quad \dots \quad \cos\left((n-1)j\frac{2\pi}{n}\right)\right]^T \tag{1}$$

$$\boldsymbol{w}_j = \left[0 \quad \sin\left(j\frac{2\pi}{n}\right) \quad \sin\left(2j\frac{2\pi}{n}\right) \quad \dots \quad \sin\left((n-1)j\frac{2\pi}{n}\right)\right]^T; \tag{2}$$

moreover, if n is even, the eigenvectors of $\lambda_{\frac{n}{2}}$ are

$$\boldsymbol{u}_{\frac{n}{2}} = [1 \ -1 \ 1 \ -1 \ \dots \ -1]^T,$$
$$\boldsymbol{w}_{\frac{n}{2}} = [0 \ \ 0 \ 0 \ \ 0 \ \ \dots \ \ 0]^T.$$

In other words, if ω is a primitive n-th root of the unity, then

$$\boldsymbol{u}_j(k) = \frac{1}{2}\left(\omega^{jk} + \omega^{-jk}\right),$$
$$\boldsymbol{w}_j(k) = \frac{1}{2\iota}\left(\omega^{jk} - \omega^{-jk}\right).$$

Note that this is an orthogonal eigenvectors basis, but with simple calculations it is possible to construct from them an orthonormal basis.

4 The Chromatic Number Problem

The *chromatic number* $\chi(G)$ is the minimum number of colors that are required to color the vertices of the graph in such a way that no two adjacent vertices have the same color. The CHROMATIC NUMBER problem consists in determining $\chi(G)$. It is well known that such a problem is NP-hard [14].

Such a problem remains NP-hard even if restricted to circulant graph. Indeed we have the following result

Theorem 1. *[9] The* CHROMATIC NUMBER *problem restricted to circulant graphs is an NP-hard problem and it is not approximable by a factor better than* $\left(\frac{n}{25}\right)^{\frac{\delta}{4}}$*, where* n *is the number of vertices of the graph.*

Here δ denote the exponent of the best approximation bound for MAXIMUM CLIQUE (i.e., MAXIMUM CLIQUE is not approximable within a factor better than n^{δ} [13], where MAXIMUM CLIQUE is the problem of finding the cardinality of subset of vertices that form a clique by having all possible edges between them).

5 Coloring Circulant Graphs

The signs of the eigenvectors associated with negative eigenvalues give useful information on correct coloring of a graph. Indeed, intuitively, we know that the value of the i-th entry of the eigenvector, multiplied by some negative value (the eigenvalue), must be equal to the sum of the entries of the eigenvector

corresponding to vertices adjacent to the given vertex i. So, if the magnitude of the eigenvalue is large enough, it is likely that such entries will have a different sign with respect to the i-th entry. This means that, by choosing a subset of eigenvectors and assigning a color to the vertex i depending on the list of signs of the i-th entries of the selected eigenvectors, we can expect an approximation to a coloring. For instance, in [2] it is shown that the signs of all eigenvectors color the graph assigning a different color to each vertex. In [1] the eigenvector information is refined algorithmically so that to obtain a correct minimum coloring with high probability. In particular, for bipartite graphs the signs of the eigenvector related to the smallest eigenvalue color the graph correctly [10].

In the case of circulant graphs, given a choice of indices $J \in \{1, \ldots, \lfloor \frac{n}{2} \rfloor\}$, the color of a vertex t will be given by the $2|J|$-dimensional vector

$$[\mathrm{sgn}(\boldsymbol{u}_j(t)), \mathrm{sgn}(\boldsymbol{w}_j(t))]_{j \in J} .$$

Indeed the following result holds:

Theorem 2. *[9] Let G be a circulant graph with n vertices, whose adjacency matrix A has nonzero elements of index $p_1 < p_2 < \ldots < p_s \leq \lfloor \frac{n}{2} \rfloor$ in the first half of the first row. Let $\boldsymbol{u}_j$ and $\boldsymbol{w}_j$ be as in (1) and (2). Let $J \in \{1, \ldots, \lfloor \frac{n}{2} \rfloor\}$ be a subset of indices such that, for all $1 \leq h \leq s$, there exists $j \in J$ for which $\boldsymbol{u}_j(p_h) < 0$. Then the signs of $\{\boldsymbol{u}_j, \boldsymbol{w}_j | j \in J\}$ color the graph correctly.*

For example, given a circulant graph with $n = 18$ and whose circulant adjacency matrix is defined by the following first row:

$$[0\,0\,1\,1\,1\,0\,0\,0\,1\,1\,1\,0\,0\,0\,1\,1\,1\,0] ,$$

let us consider the sign patterns of $\boldsymbol{u}_3$ and $\boldsymbol{w}_3$, that are

$$[+\,+\,-\,-\,-\,+\,+\,+\,-\,-\,-\,+\,+\,+\,-\,-\,-\,+]$$

and

$$[+\,+\,+\,+\,-\,-\,+\,+\,+\,+\,-\,-\,+\,+\,+\,+\,-\,-] ,$$

respectively. By Theorem 2 we have that $\{\boldsymbol{u}_3, \boldsymbol{w}_3\}$ color the graph correctly. Namely, taking $R = (+,+)$, $Y = (+,-)$, $B = (-,+)$, and $G = (-,-)$, we obtain the following correct coloring (see Figure 1):

$$[R\ R\ B\ B\ G\ Y\ R\ R\ B\ B\ G\ Y\ R\ R\ B\ B\ G\ Y].$$

In the case of circulant graphs of degree 2 a pair of eigenvectors associated with negative eigenvalues color the graph correctly. It is interesting to note that in the case of bipartite circulant graph the eigenvector that colors the graph correctly is $\boldsymbol{u}_{\frac{n}{2}}$. Moreover, the following result holds:

Theorem 3. *[9] Given a circulant graph of degree 3 or 4, there exists an integer i $(0 < i < \lfloor \frac{n}{2} \rfloor)$ such that $\boldsymbol{u}_i$ and $\boldsymbol{w}_i$ color the graph correctly, unless the graph is made by cliques of order 5.*

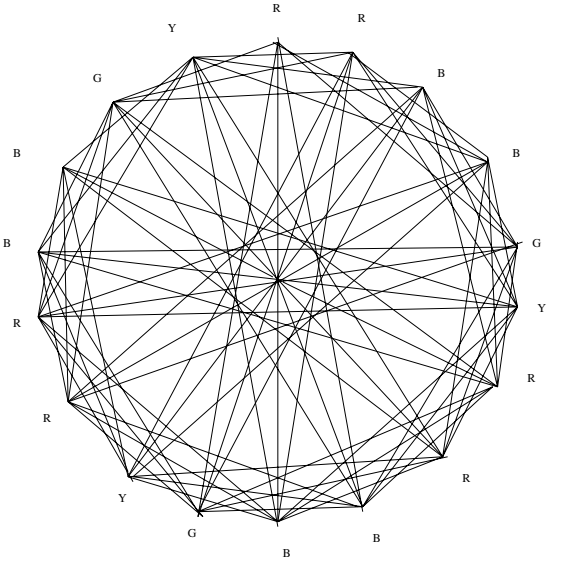


Fig. 1. A circulant graph colored by sign pattern of its eigenvectors.

In the case of graphs composed by cliques of order 5, we have that $\boldsymbol{u}_1$, $\boldsymbol{w}_1$ and $\boldsymbol{u}_2$ color the graph correctly. Furthermore we have that

Theorem 4. *[9] Let G be a circulant graph whose adjacency matrix A satisfies*

$$p_{j+1} = p_1 + j \qquad 0 \leq j \leq \left\lfloor \frac{n}{2} \right\rfloor - p_1 \ .$$

Then, $\lfloor \frac{n}{2p_1} \rfloor$ eigenvectors correctly color the graph.

6 Proposed Heuristics

The problem we investigate in this section is to find the minimum number of eigenvectors that color a circulant graph correctly. To this goal, in this paper we propose two different heuristic algorithms and, to test their efficiency we compare them with the Johnson algorithm [3]. The Johnson algorithm is one of the most simple algorithm to color a graph, but in the average case it always gives good performances. Let $G = \langle V, E \rangle$ be a graph and $N(v)$ be the set of adjacent vertices to the vertex v. The Johnson algorithm is the following:

$x \leftarrow 1$;
$W \leftarrow V$ (where W is the set of non-colored vertices);
while $W \neq \varnothing$ do
    $U \leftarrow W$;
    while $U \neq \varnothing$ do
        let v be the vertex with smallest degree in the subgraph induced from U;
        color v with x;

$U \leftarrow U - \{v\} - \{N(v)\}$;
$W \leftarrow W - \{v\}$;
end
$x \leftarrow x + 1$;
end

By our first heuristic we select at each step the eigenvector that covers the largest number of the entries equal to 1 in the first row of the adjacency matrix $A(G)$. If there are more than one of such eigenvectors, we choose the one that requires to add the minimum number of colors in the previous obtained coloring.

When we have covered all the 1 entries of the first row of $A(G)$, we give a color to each vertices by the sign pattern of the selected eigenvectors. Moreover, to reduce the number of color, we construct a graph $\hat{G} = \langle \hat{V}, \hat{E} \rangle$ using the following procedure:

- each vertex $\hat{v} \in \hat{V}$ corresponds a class of color in G;
- for each an edge in E, between two vertices $v_1 \in V$ and $v_2 \in V$, we construct an edge in $\hat{E}$, between two vertices $\hat{v_1} \in \hat{V}$ and $\hat{v_2} \in \hat{V}$ associated to the color class of v_1 and v_2.

Then we color the graph $\hat{G}$ by the Johnson algorithm. So we color each vertex in G with the color given in the corresponding class of color in $\hat{G}$.

Let $P = \{p_i | p_i \leq \lfloor \frac{n}{2} \rfloor$ and p_i is an index of a nonzero elements in the first half of the first row of $A(G)\}$.

Our first heuristic is the following:

$Q \leftarrow P$ *(where* P *is the set of non-covered indices);*
while $Q \neq \varnothing$ *do*
let $\boldsymbol{u}_i$ *be the eigenvector whose negative components cover the largest number of elements in* Q*;*
if $\boldsymbol{u}_i$ *is not unique choose the one that requires to add the minimum number of colors in the previous obtained coloring;*
compute Q*;*
end
compute the graph $\hat{G}$*;*
color $\hat{G}$ *by the Johnson algorithm;*
color G *by the coloring of* $\hat{G}$*;*

The second heuristic is quite similar to the first one but now at each step we select a couple of eigenvectors of the type (1) instead of a single eigenvector. We further check if both of the eigenvectors are necessary to the coloring, then we use just one of them.

The second proposed heuristic is the following:

$Q \leftarrow P$ *(where* P *is the set of non-covered indices);*
while $Q \neq \varnothing$ *do*
let $\{\boldsymbol{u}_i, \boldsymbol{u}_j\}$ *be the eigenvectors whose negative components cover the largest number of elements in* Q*;*
if $\boldsymbol{u}_i$ *and* $\boldsymbol{u}_j$ *are not both necessary to the coloring select the necessary one;*

Table 1. Results obtained by the first heuristic; the ones better than the Johnson algorithm are in bold.

d\n	50	60	80	100	200	300	400	500
3	3.01	3.93	4.00	4.00	4.00	3.98	4.00	4.00
5	3.78	4.07	4.09	4.07	4.03	4.03	4.04	4.02
8	**4.63**	**4.04**	**4.01**	**4.08**	**3.88**	**3.88**	**3.87**	**3.88**
9	6.10	7.35	6.34	5.72	4.57	4.43	4.33	4.23
10	6.06	4.95	**4.73**	**4.76**	**4.05**	3.99	**3.97**	**3.95**
12	8.96	6.61	6.77	6.45	**4.88**	**4.42**	**4.19**	**4.14**
15	8.93	8.61	8.98	8.52	7.45	7.03	8.39	7.74
20	10.86	9.48	10.47	10.32	9.96	9.87	11.33	10.85
30	15.14	13.19	13.36	13.53	13.35	12.57	13.22	13.53
40	21.85	18.56	17.25	17.07	15.32	14.40	14.76	14.72
50		25.02	21.66	21.81	19.69	17.37	16.70	15.11

Table 2. Results obtained by the second heuristic; the ones better than the Johnson algorithm are in bold.

d\n	50	60	80	100	200	300	400	500
3	3.01	3.93	4.00	4.00	4.00	3.98	4.00	4.00
5	3.78	4.11	4.13	4.09	**4.01**	4.03	**4.02**	4.01
8	**4.53**	**4.08**	**4.12**	**4.04**	**3.89**	**3.84**	**3.90**	**3.89**
9	5.74	5.50	5.30	**4.97**	**4.40**	**4.34**	**4.26**	**4.17**
10	5.88	**4.87**	**5.05**	**4.83**	**4.12**	**3.95**	**3.98**	**3.97**
12	7.01	5.84	6.25	6.00	**4.94**	**4.44**	**4.34**	**4.22**
15	8.63	8.52	8.59	7.88	7.57	**6.75**	7.11	**6.70**
20	10.33	8.94	9.62	8.96	8.28	8.67	9.15	9.11
30	14.69	12.92	13.17	12.82	11.62	**10.06**	10.63	**10.74**
40	21.69	17.86	17.11	16.70	14.48	12.88	13.50	**12.73**
50		24.98	21.74	21.56	19.38	16.07	15.69	**14.80**

Table 3. Results obtained by the Johnson algorithm.

d\n	50	60	80	100	200	300	400	500
3	2.89	3.25	3.56	3.33	3.59	3.26	3.70	3.45
5	3.62	3.83	3.91	3.89	4.01	3.97	4.15	3.99
8	4.77	4.23	4.45	4.45	4.55	4.49	4.72	4.70
9	5.31	5.10	5.08	5.19	4.98	5.15	5.17	5.27
10	5.62	4.90	5.22	5.16	5.06	4.99	4.97	5.16
12	6.08	5.80	6.01	5.86	5.86	5.56	5.62	5.64
15	7.36	7.37	7.20	7.00	6.76	6.88	6.78	6.71
20	8.44	8.25	8.28	8.21	7.99	7.80	7.93	7.95
30	13.46	12.18	11.59	11.04	10.40	10.40	10.13	10.64
40	20.85	16.38	14.63	13.96	13.26	12.83	13.07	12.92
50		23.02	19.12	16.83	15.48	15.18	15.04	14.80

if $\{\boldsymbol{u}_i, \boldsymbol{u}_j\}$ *is not unique choose the one that requires to add the minimum number of colors in the previous obtained coloring;*
compute Q*;*
end
compute the graph $\hat{G}$*;*
color $\hat{G}$ *by the Johnson algorithm;*
color G *by the coloring of* $\hat{G}$*;*

7 Experimental Results

To test the efficiency of our heuristics we have implemented these algorithms and the Johnson one in the C language on a serial computer. We have considered graphs with a number of vertices n between 50 and 500, and with a degree d between 3 and 50. For each pair (n, d) we have generated 1000 circulant graphs in a random way. Then we have computed the average of colors used for each pair for the three algorithms.

The results obtained by the first heuristic are shown in Table 1.

By the second heuristic we obtained the results given in Table 2. Finally, in Table 3 we report the number of colors obtained with the Johnson algorithm.

In the 78.2% of the cases, the results of the second heuristic are better than the ones obtained by the first one. Comparing the second heuristic with the Johnson algorithm, we notice that, just in the 36.8% of the cases, the results of this heuristic are better than the ones given by the classical one. Let consider now the gray area in table 2, that corresponds to the case of sparse graphs with degree greater than 5. The performance of the second heuristic becomes better in the 75% of the results. Note that for graphs with degree lower or equal to 5, spectral heuristics quite often use 2 eigeinvectors to color the graph which determine 4 different colors. Thus, in this case the results are not better than Johnson algorithm.

A further experimentation could consist of selecting, instead of couple of eigenvectors, larger groups of them. In this case the results could be still better.

References

1. Alon, N., Kahale, N.: A Spectral Technique for Coloring Random 3-Colorable Graphs. Proceedings of The 26th Annual Symposium on the Theory of Computing. ACM Press, New York. (1994) 346–355.
2. Aspvall, B., Gilbert, J. R.: Graph Coloring using Eigenvalue Decomposition. Algebraic Discrete Methods. **5** (1984) 526–538.
3. Berger, B., Rompel, J.: A Better Performance Guarantee for Approximate Graph Coloring. Algorithmica. **5** (1990) 459–466.
4. Bermond, J.C., Comellas, F., Hsu, D.F.: Distributed loop computer networks A survey. Journal of Parallel andDistributed Computing **24** (1995) 2–10.
5. Biggs, N.: Algebraic Graph Theory. Cambridge University. Press, Cambridge. (1974).

6. Burkard, R.E., Sandlholzer, W.: Efficiently solvable special cases of bottleneck travelling salesman problems. Discrete Applied Mathematics. **32** (1991) 61–76.
7. Chan, T.: An Optimal Circulant Preconditioner for Toeplitz System. SIAM J. Sci. Stat. Comput. **9** (1988) 766–771.
8. Chan, R., Yip, A. M., Ng, M.K.: The best Circulant Preconditioners for Hermitian Toeplitz matrices. SIAM J. Numeric. Anal. **38** (2001) 876–896.
9. Codenotti, B., Gerace, I., Vigna, S.: Hardness Results and Spectral Techniques for Combinatorial Problems on Circulant Graphs. Linear Algebra and its Applications. **285** (1998) 123–142.
10. Cvectović, D. M., Doob, M., Sachs H.: Spectra of Graphs. Academic Press, New York. (1978).
11. Elspas, B., Turner, J.: Graphs with Circulant Adjacency Matrices. J. of Combinatorial Theory **9** (1970) 297–307.
12. Gerace, I., Pucci, P., Ceccarelli, N., Discepoli, M., Mariani, R.: A Preconditioned Finite Elements Method for the p-Laplacian Parabolic Equation. Appl. Num. Anal. Comp. Math. **1** (2004) 155–164.
13. Håstad, J.: Clique is Hard to Approximate within n to the power (1-epsilon). Acta Mathematica **182** (1999) 105–142.
14. Karp, R. M.: Reducibility among Combinatorial Problems. R. E. Miller and Thatcher (eds.), Complexity of Computer Computations, Plenum Press, New York. (1972) 85–103.
15. Leighton, F.T.: Introduction to parallel algorithms and architecture: Arrays, trees, hypercubes. M. Kaufman (1992).
16. Litow, B., Maus, B.: On isomorphic chordal ring. Proc. of The Seventh Australian Workshop on Combinatorial Algorithms (AWOCA'96), BDCS-TR-508 (1996) 108–111.
17. Lund, C., Yannakakis M.: On the hardness of approximating minimization problems. Journal of the ACM **41** (1994) 960–981.
18. Turner, J.: Point-Symmetric Graphs with Prime Number of Points. Journal of Combinatorial Theory. **3** (1967) 136–145.

A Method to Establish the Cooling Scheme in Simulated Annealing Like Algorithms

Héctor Sanvicente-Sánchez[1] and Juan Frausto-Solís[2]

[1] IMTA, Paseo Cuauhnáhuac 8532, Col Progreso, C.P. 62550, Jiutepec, Morelos, México
hsanvice@tlaloc.imta.mx
[2] ITESM Campus Cuernavaca, Reforma 182-A, Col. Lomas de Cuernavaca, A.P. 99-C, Cuernavaca, Morelos, México
jfrausto@campus.mor.itesm.mx

Abstract. Since the publication of the seed paper of Simulated Annealing algorithm (SA) written by Kirkpatrich, several methods have been proposed to get the cooling scheme parameters. Although developed for SA, some of these methods can be extended to the algorithm known as Threshold Accepting (TA). SA and TA are quite similar and both are treated in this paper as a Simulated Annealing Like (SAL) algorithm. This paper presents a method to set the cooling scheme parameters in SAL algorithms; it establishes that both, the initial and the final temperatures are function of the maximum and minimum cost increment getting from the neighborhood structure. Experimentation with Traveling Salesman Problem and Hydraulic Network Design Problem shows that the cooling schemes getting through our method are more efficient than the previous ones.

Keywords: Simulated Annealing, Threshold Accepting, Combinatorial Optimization and Heuristic Optimization, Simulated Annealing Like Algorithms.

1 Introduction

A Simulated Annealing Like (SAL) algorithm [1] is any one that works with a Simulated Annealing (SA) approach. The classical SA of Kirkpatrick [2] and Threshold Accepting (TA) [3] among many others can be classified in this category.

SA [2] is a simple and effective optimization method to find near optimal solutions to NP-hard combinatorial problems [4]. A SA algorithm may be seen like a Markov chain sequence [5] (an homogeneous one); where L_k identifies the length of each Markov chain and it must be $L_k > 0$ (k is the sequence index). The states in a Markov chain are established by the solution space S of the optimization problem. The sequence of Markov chains is built on a descending sequence of a control parameter c_k, commonly referred as the temperature ($c_k > 0$). The output of a Markov chain is the solution $S_{eq}^{\ k} \in S$, where $S_{eq}^{\ k}$ is a solution getting when the dynamic equilibrium or the stationary distribution is reached. The control parameter must satisfy the following property:

$$\lim_{k\to\infty} c_k = 0$$

$$c_k \geq c_{k+1} \qquad \forall\, k \geq 1$$

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 755–763, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Consecutive temperatures c_k's are setting through a cooling function:

$$c_{k+1} = f(c_k)$$

In this way, the SA did a stochastic walk on the solution space of an optimization problem. The stochastic walk for each Markov chain is done until the stationary distribution is reached and, it depend on the temperature parameter. During the stochastic walk, cost deteriorations are accepted with a probability, it is known that the accepting probability is decreasing along the iterations.

Similarly, TA [3] did a stochastic walk on the solution space of an optimization problem. It also uses a cooling scheme to control the transition probabilities among solutions in order to accept solutions with cost deterioration. The distribution of probabilities (usually Boltzmann in SA) is modeled in TA through a hidden distribution in a parameter known as the threshold.

Asymptotic convergence to the optimal solution is one of the main features of SAL algorithms [5]; for this reason SAL algorithms are considered like approximation algorithms [6,7]. So a balance between efficiency and efficacy need to be reach.

Since the publication of the seed paper of SA algorithm [2], several methods and procedures have been proposed to reduce the executing time of SA. Most of these methods have been focused to the cooling scheme parameters [6, 7]. The cooling scheme gives a natural and intuitive way of controlling the executing time and it establishes the balance between efficiency and efficacy. Although developed for SA, some of these methods can be extended to SAL algorithms. However, these methods are based on experimentation and a tuning process requiring a lot of time and effort. Therefore, for this tuning process, a method to determine the scheme parameters with a reduced experimentation or without experimentation at all would be very advantageous.

This paper present a method to determine cooling scheme parameters in SAL algorithms focused in SA and TA. The method has been tested with SA implementations to solve several instances of two NP-hard problems: the Traveling Salesman Problem (TSP) and the Hydraulic Networks Design Problem (HNDP).

2 Initial and Final Temperatures

The temperature parameter has as extreme bounds of initial and final temperature. These bounds establish the extreme accepting probabilities during an executing SAL algorithm.

The initial temperature c_0 should be set at a value in such a way that all the transitions in the Markov chain at c_0 be accepted. It is, this temperature should not constrain a free movement of the search procedure on the solution space. However, if it is set too high, a lot of time is spending at the beginning of the process; on other hand, if it is too low, the search procedure will be trapped in a local optimum. In general, the initial temperature value is set through an iterative tuning procedure and it has an initial iterative solution dependency [1, 2, 5, 6, 7, 8].

The final temperature c_f establishes the stop criteria in a SAL algorithm. Then, in a similar way to c_0, if it is set too high the final solution will be trapped in a local optimum, but if it is set too low the SAL algorithm spend a lot of time at the ending of the process. The most common ways of determine this parameter is [1, 2, 5, 6, 7, 8]:

a) setting it close to zero, b) setting a number of unchanging temperature cycles and c) through an adaptive method (using the mean and standard deviation parameters). The first two options can be implemented in an arbitrary way or through a tuning procedure. Adaptive methods may produce a premature ending of the process because the mean and the standard deviation lose variability along the time.

To avoid disadvantages of the above methods, a new way of setting the bound temperatures is developed below. The calculation of the extreme bounds for the temperature parameter is based on the following considerations:

Let $P^A(S_j)$ the accepting probability of a proposed solution S_j generated from the actual solution S_i and $P^R(S_j)$ the rejecting probability, then:

$$P^R(S_j) = 1 - P^A(S_j)$$

Accepting or rejecting a proposed solution depends on cost deterioration size that will be produced to the actual solution:

$$P^A(S_j) = g(Z(S_i) - Z(S_j)) = g(\Delta Z_{ij})$$

$Z(S_i)$ is the cost function (objective function) of the optimization problem and $g(\Delta Z_{ij})$ is a probability distribution function which gives a probability for the cost difference $\Delta Z_{ij} = Z(S_i) - Z(S_j)$.

Let the neighborhood of a solution S_i defined by:

$$\{\forall\ S_i \in S, \exists \text{ a set } V_{Si} \subset S \,/\, V_{Si} = V: S \rightarrow S\ \}$$

where V_{Si} is the neighborhood set of S_i and $V: S \rightarrow S$ is a mapping. Then the neighbors of a solution S_i depend on the neighborhood structure V established.

The maximum and minimum deteriorations produced through the neighborhood structure V are:

$$\Delta Z_{Vmax} = \text{Max}\{Z(S_j) - Z(S_i)\} \qquad \forall\ S_j \in V_{Si}, \forall\ S_i \in S$$

$$\Delta Z_{Vmin} = \text{Min}\{Z(S_j) - Z(S_i)\} \qquad \forall\ S_j \in V_{Si}, \forall\ S_i \in S$$

Where S_i is a solution and V_{Si} the neighborhood of it. The neighborhood structure can be established in different ways, for example a neighbor of S_i could be any solution that is different of S_i in just one item. In this case, ΔZ_{Vmax} (ΔZ_{Vmin}) could be setting by getting the maximum (minimum) cost produced changing on S_i just one item.

ΔZ_{Vmax} and ΔZ_{Vmin} give the maximum and minimum deteriorations that may be produced during the execution of a SAL algorithm.

Now, the initial temperature in a SAL algorithm should permit free movements in a search procedure, satisfying $P^A(S_j) = P^A(\Delta Z_{ij}) \approx 1$.

Like ΔZ_{Vmax} gives the maximum deterioration that may be produced during the execution of a SAL algorithm. The way of make sure that ΔZ_{Vmax} be accepted, at the initial temperature c_0, is setting the acceptation probability with $P^A(\Delta Z_{Vmax}) \cong 1$ and calculates c_0 like:

For SA: $$c_0 = -\Delta Z_{Vmax} / \ln(P^A(\Delta Z_{Vmax})) \qquad (1)$$

where $P^A(\Delta Z_{Vmax})$ may be 0.90, 0.95, 0.99.

For TA: $$c_0 = \Delta Z_{Vmax} \qquad (2)$$

The above equation makes sure that for TA any deterioration at c_0 will be accepted.

The final temperature in a SAL algorithm should be setting in such a way that: 1) there are not yet deteriorations in the cost value or 2) the probability to accept deteriorations is too low. These two conditions can be written as $P^A(\Delta Z_{ij}) \approx 0$.

ΔZ_{Vmin} establishes the minimum deterioration that can be produced during the execution of a SAL algorithm. In a similar way that for c_0 temperature, the final temperature c_f is setting by the next equations:

$$\text{For SA:} \qquad c_f = -\Delta Z_{Vmin} / \ln(P^A(\Delta Z_{Vmin})) \tag{3}$$

where $P^A(\Delta Z_{Vmin})$ may be 0.10, 0.05, 0.01.

$$\text{For TA:} \qquad c_f \leq \Delta Z_{Vmax} \tag{4}$$

The above way of setting c_f temperature makes able to have a control of the accepting probability for low temperatures and, in an indirect way, we can control the climbing probability of a local optimum.

This way of determining the initial and final temperatures c_0 and c_f are totally equivalent for SA and TA. The method reduces tuning time, getting a c_0 independently of an initial solution. Besides a final temperature c_f is established from a strong stop criteria instead of an arbitrary one.

3 Markov Chain Length and Cooling Function

SAL algorithms can be modeled using homogeneous Markov chains. This model establishes that for each Markov chain, the stationary distribution must be reached; that is, for each temperature value c_k the length of the Markov chain L_k must be set to restore the stochastic equilibrium. There exists a hard relation between Markov chain length and the cooling speed; while bigger is the cooling step, longer is the Markov chain to restore the equilibrium, but for small cooling steps, small Markov chain length are necessary. However, like $L_k \rightarrow \infty$ for $c_k \rightarrow 0$ [5, 8], then L_k must be bounded to avoid extremely long Markov chains for small values of c_k.

There are two main criteria to set L_k [1, 5, 8]: a) make L_k constant, and b) through and adaptive criteria. For the first case, some examples can be $L_k = n$, the number of problem's variables [5], or $L_k = m\,|V_{Si}|$, a neighborhood size multiple [5, 8]. For the second case, an adaptive criteria is established when the mean and standard deviation parameters are stabilized. Like the adaptive methods may produce $L_k \rightarrow 0$ instead of $L_k \rightarrow \infty$, because the mean and the standard deviation lose variability as $c_k \rightarrow 0$, we proposed to use $L_k = L = m\,|V_{Si}|$ based on the next analysis.

As a SAL algorithm is implemented through a neighborhood structure, then the maximum number of solutions rejected from an actual solution S_i is the neighborhood size $|V_{Si}|$. In this way for a SAL algorithm that has reached a value of c_k in which $P^A(\Delta Z_{ij}) \approx 0$, the maximum Markov chain length Lmax can be established like the maximum number of solutions evaluated and rejected when the optimum has been reached. Then:

$$L_k = L \leq Lmax = g(|V_{Si}|)$$

Two sampling methods to explore the neighborhood can be implemented in a SAL algorithm: a) replacement sampling and b) non-replacement sampling. For a non-replacement sampling, Lmax is established as:

$$Lmax = |V_{Si}|$$

Then, the value of L_k depend on the number of different neighbor elements that must be explored at the lowest temperature, that is:

$$L_k = L \leq |V_{Si}| \tag{5}$$

It is known that for a replacement sampling, the number of different elements that can be gotten in N samples depends on the neighborhood sampling distribution. For instance, in the case of an uniform neighborhood sampling distribution given by:

$$G(c_k) = G = 1/|V_{Si}| \qquad \forall\, S_j \in V_{Si}$$
$$= 0 \qquad \forall\, S_j \notin V_{Si}$$

It is also known that the expected fraction of different elements of V_{Si} that are selected by N samples with replacement is equal to the probability $p(S_j)$ of selecting a given element S_j from V_{Si} in N samplings [5]:

$$p(S_j) = 1 - e^{-(N / |VSi|)}$$

Then, the number N of samplings that must be taken from the neighborhood V_{Si}, in order to expect a fraction $p(S_j)$ of different elements is:

$$N = -|V_{Si}| \ln(1 - p(S_j))$$

If $N = C\,|V_{Si}|$, then $C = -\ln(1 - p(S_j))$. With a sampling rate of 99%, $N = 4.6\,|V_{Si}|$. With a sampling of $N = 3\,|V_{Si}|$ we get $p(S_j) = 0.95$, that is 95% of the neighbors are sampled. With $N = 2\,|V_{Si}|$, we have $p(S_j) = 0.86$, and for $N = |V_{Si}|$, we have $p(S_j) = 0.63$.

For a SAL algorithm using a random neighborhood sampling with replacement, the Markov chain length must be set as:

$$L_k = L = N = C\,|V_{Si}| \tag{6}$$

where $1 \leq C \leq 4.6$, to make sure that the Markov chain building at the lowest temperature evaluates an adequate fraction of different neighbor elements from the actual solution. Then $L_k = L \leq Lmax$ depends on the neighborhood-exploring rate that we want at the lowest temperature (Aarts and Korst [5] made C = 1).

The most common used cooling function in SAL (and used in this work) is the geometric reduction function proposed by Kirkpatrick [2]:

$$c_{k+1} = \alpha\, c_k \tag{7}$$

where $\alpha \approx 1$, it is normally in the range of $0.7 \leq \alpha \leq 0.99$.

The selection of this function is based on the annealing analogy where convergence to the optimal solution depends on cooling speed. The geometric reduction function is an easy an intuitive way of setting cooling speed. Cooling speed is faster when $\alpha \rightarrow 1$. Then, fixing the rest of cooling scheme parameters (initial temperature c_0, final temperature c_f and Markov chain length, $L_k = L$) the algorithm precision and efficacy are controlled through α tuning.

4 Experimental Results

To test our method to set the cooling scheme, we develop two SA algorithms to solve the next NP-hard problems Traveling Salesman Problem (TSP) and Hydraulic Network Design Problem (HNDP), and compare its performance with a tuning method.

Traveling Salesman Problem (TSP) consists on to find the shortest distance tour through a set of n cities, crossing every city exactly once. It assumes that always there is a path joining two towns directly. HNDP consists on to set the pipelines diameters of a hydraulic network to satisfy the demands in flow and pressure with a minimal building cost.

For each problem two instances were solved, gr120 and si1032 for TSP and AS and CEA for HNDP. TSP instances were taken from the TSPLIB [9] and they have 120 and 1032 towns, respectively. HNDP instances were taken from [10] and [11] respectively. The AS is a network with 7 nodes, 8 pipelines, 1 water source, 2 cycles and catalogs of 4 or 5 diameters for each pipeline. The CEA is a network with 7 nodes, 9 pipelines, 1 water source, 3 cycles and a general catalog of 6 diameters.

Four cooling schemes were setting to solve each instance, two for a tuning experimental method, and two for our analytical method. The main difference between both methods is the way of setting the c_0 and c_f temperatures. The difference between cooling schemes for a same method is the cooling speed and was doing like a way of tuning the algorithm precision.

- The Markov chain length L was setting for exploring 86% of the neighborhood size at the lowest temperature. It is after $2|V_{Si}|$ iteration.
- From the cooling function $c_{k+1} = \alpha c_k$, the two cooling speed used to establish the two cooling schemes for each instance are $\alpha = 0.85$ and $\alpha = 0.95$.

For the tuning method c_0 and c_f were setting like:

- The initial temperature c_0 was established through the following iterative tuning procedure: Choose a big c_0 temperature and does a number of iterations, if the accepting rate, defined like the number of transitions accepted with respect of the total proposed transitions, is less than a value X_0, double c_0 and does the number of iterations again. Process continues until the accepting rate is greater than X_0.
- The final temperature c_f was defined through a threshold parameter close to zero.

For our cooling scheme c_0 and c_f were setting like:

- Exploring the cost data arrays through the neighborhood structure we found ΔZ_{Vmax} and ΔZ_{Vmin} and using (1) and (3) with $P^A(\Delta Z_{Vmax}) = 0.95$ and $P^A(\Delta Z_{Vmin}) = 0.05$ the values of c_0 and c_f were calculated.

The TSP-SA and HNDP-SA algorithms were tested running them on a Silicon Graphics workstation O2-R10000 at 200 Mhz. Table 1 gives the cooling schemes used to test the TSP-SA algorithm and Table 3 shows the cooling schemes employed to solve the HNDP instances.

From table 1 and table 3, it can be notice that c_0 values getting through our method are lower than the values getting by tuning, and c_f values are higher than the tuning values, while α and L remain the same. We did 20 runs for each cooling scheme and

Table 1. Cooling schemes used to solve the gr120 and si1032 TSP instances.

Method	Instance	α	c_0	c_f	L
Tuning	gr120	0.85	3114252	0.05	28800
Tuning	gr120	0.95	3114252	0.05	28800
Tuning	si1032	0.85	511171	0.05	2130048
Tuning	si1032	0.95	511171	0.05	2130048
Our method	gr120	0.85	92156	0.33	28800
Our method	gr120	0.95	92156	0.33	28800
Our method	si1032	0.85	35677	0.66	2130048
Our method	si1032	0.95	35677	0.66	2130048

Table 2. TSP-SA algorithm performance.

Method	Instance	α	Mean cost	Max. cost	Min. cost	Mean time
Tuning	gr120	0.85	8200	8800	7800	6.9
Tuning	gr120	0.95	8200	8500	7800	22
Tuning	si1032	0.85	110200	110600	109000	1240
Tuning	si1032	0.95	10800	110600	106000	3954
Our method	gr120	0.85	8200	8600	7800	2.7
Our method	gr120	0.95	7900	8200	7800	8.8
Our method	si1032	0.85	109000	110500	108000	690
Our method	si1032	0.95	107500	108000	106000	2200

Table 3. Cooling schemes used to solve the AS and CEA HNDP instances.

Method	Instance	α	c_0	c_f	L
Tuning	AS	0.85	6800000	0.001	80
Tuning	AS	0.95	6800000	0.001	80
Tuning	CEA	0.85	786365	0.001	108
Tuning	CEA	0.95	786365	0.001	108
Our method	AS	0.85	2339487	1001	80
Our method	AS	0.95	2339487	1001	80
Our method	CEA	0.85	1831352	1991	108
Our method	CEA	0.95	1831352	1991	108

Table 4. HNDP-SA algorithm performance

Method	Instance	α	Mean cost	Max cost	Min cost	Mean time
Tuning	AS	0.85	448000	455000	444000	9.45
Tuning	AS	0.95	445350	448000	444000	29.82
Tuning	CEA	0.85	361915.31	369890.17	354605.74	9.75
Tuning	CEA	0.95	357661.92	367161.4	354605.74	31.28
Our method	AS	0.85	447200	456000	444000	1.3
Our method	AS	0.95	445500	448000	444000	4.2
Our method	CEA	0.85	361868.56	373142	357700.58	2.68
Our method	CEA	0.95	356146.76	358710.86	354605.74	8.92

the performance getting are showed at table 2 and table 4 for each NP-hard problem, respectively.

Table 2, shows that sometimes the performance of the TSP-SA algorithm through our cooling scheme versus the tuning experimental one got similar costs. Notice however, that in some cases the costs gotten through our cooling scheme are better. This table shows also, that the executing time (mean time) of our method is better than the experimental tuning with savings among 40% and 66%.

Table 4 shows that the performance of the HNDP-SA algorithm through our cooling scheme got in general better costs than through the tuning experimental cooling scheme. Besides, the executing times of our method have savings around 70% than the experimental one.

5 Conclusions

In this paper we have presented a new method to set the cooling scheme that can be applied to any Simulated Annealing Like (SAL) algorithm. This method is based on the accepting distribution function used in each algorithm in order to set the c_0 and c_f temperatures. It is considered Threshold Accepting (TA) and the classical Simulated Annealing (SA) algorithms as SAL algorithms. We consider TA has a hidden accepting distribution function given through the threshold parameter. The method is able to avoid the over-heating given by a huge c_0 value and the under-heating produced by the initial solution dependency in an iterative process. Besides, it avoids the over-freezing given by the arbitrary setting of c_f close to zero, and the quenching produced by a faster stop criterion. In general the method simplifies the process of the cooling scheme through the establishment of parameter bounds chosen from a theoretically basis. The SA cooling schemes gotten throughout our method are more efficient than cooling schemes gotten with tuning experimentation because we always save tuning time. We have tested our cooling scheme with two NP-hard problems using SAL algorithms; the experimentation shows that the set of solutions for these problems gotten with our method has always the same precision, but in general the results are gotten faster.

References

1. Sanvicente-Sánchez, H., 2003. Metodología de paralelización del ciclo de temperaturas en algoritmos tipo recocido simulado. Tesis Doctoral. Instituto Tecnológico y de Estudios Superiores de Monterrey (Campus Cuernavaca), México, 255 pp.
2. Kirkpatrik, S., Gelatt Jr., C.D. and Vecchi, M.P., 1983. Optimization by simulated annealing. Science, Vol. 220, No. 4598, pp. 671–680.
3. Dueck, G. and Scheuer, T., 1990. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. Journal of Computational Physics, No. 90, pp. 161–175.
4. Papadimitriou, C.H., 1994. Computational complexity. Addiso-Wesley Publishing Company, USA, 523 pp.

5. Aarts, E. and Korst, J., 1989. Simulated annealing and Boltzmann machines: A stochastic approach to combinatorial optimization and neural computing. John Wiley & Sons, Great Britain, 272 pp.
6. Sanvicente-Sánchez, H., 1997. Recocido simulado: optimización combinatoria. Estado del arte. Instituto Tecnológico y de Estudios Superiores de Monterrey (Campus Cuernavaca), México. 72 pp.
7. Sanvicente-Sanchez, H., 1998. Recocido simulado paralelo. Propuesta de Tesis Doctoral. Instituto Tecnológico y de Estudios Superiores de Monterrey (Campus Cuernavaca), México, 38 pp.
8. Dowsland, K.A., 1993. Simulated annealing. In: C.R. Reeves (Editor): Modern heuristic techniques for combinatorial problems. John Wiley and Sons, Great Britain, pp. 20–69.
9. Reinelt, G., 1995. TSPLIB95. http://softlib.rice.edu/softlib/tsplib.
10. Alperovits, E. and Shamir, U., 1977. Design of optimal water distribution systems. Water Resources Research, Vol. 13, No, 6, pp. 885–900.
11. Carrillo, S.J.J, Islas, M.U., Gómez, B.H.A. y Vega, S.B.E., 1998. Selección de las tuberías de una red de distribución de agua potable para que sea eficiente y económica. XVIII Congreso Latinoamericano de Hidráulica, Oaxaca, Oax., México, 13 – 16 octubre, pp. 719–728.

Packing: Scheduling, Embedding, and Approximating Metrics*

Hu Zhang

Institute of Computer Science and Applied Mathematics, University of Kiel,
Olshausenstraße 40, D-24098, Kiel, Germany
hzh@informatik.uni-kiel.de

Abstract. Many problems in computer science are related to scheduling problems or embedding problems. Therefore it is an interesting topic to find efficient (approximation) algorithms for these two classes of problems. In this paper, we present fast approximation algorithms for scheduling on unrelated machines, job shop scheduling, network embeddings and approximating metrics. As the usual technique, we consider the fractional relaxations of the integral problems. By appropriate formulation, these problems can be solved with the approximation algorithms for general packing problems by Jansen and Zhang [18] (with rounding techniques if necessary). For approximating metrics problem, which can not be solved by most traditional methods efficiently, we show that our algorithm can deliver the desired approximate solution fast.

1 Introduction

Scheduling problems can find many interesting applications in computer science. Moveover, many problems of networks with general structure are hard. One approach is to approximate the complicated networks by simple ones. Thus the embedding problems arise. In this paper we will study some problems in the above two classes which can be formulated as the packing problem. In addition, in real applications reducing the running times of the algorithms is very important, especially for real-time systems. Therefore here we concentrate on this issue.

The general *packing problem* (or *convex min-max resource-sharing problem*), which includes many important problems, is defined as follows:

$$(P) \quad \text{compute } x^* \in B \text{ such that } \lambda^* = \lambda(x^*) = \min\{\lambda | f(x) \leq \lambda \cdot \mathbb{1}, x \in B\},$$

where $f : B \to \mathbb{R}_+^M$ is a vector of M continuous convex functions defined on a nonempty convex compact set $B \in \mathbb{R}^N$, and $\mathbb{1}$ is the vector of all ones. The functions f_m, $1 \leq m \leq M$, are the packing constraints. In addition we define $\lambda(x) = \max_{1 \leq m \leq M} f_m(x)$ for any fixed $x = (x_1, \ldots, x_N) \in B$.

* This research was supported in part by the DFG Graduiertenkolleg 357, Effiziente Algorithmen und Mehrskalenmethoden, by EU Thematic Network APPOL, Approximation and Online Algorithms for Optimization Problems, IST-2001-32007, and by EU Project CRESCCO, Critical Resource Sharing for Cooperation in Complex Systems, IST-2001-33135.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 764–775, 2004.
© Springer-Verlag Berlin Heidelberg 2004

There are many applications of the general packing problem. Typical examples include the Held-Karp bound for TSP, minimum-cost multicommodity flows, maximum concurrent flow, bin covering, spreading metrics, graph partitioning, and multicast congestion in communication networks [3,9,11,14,17,19, 23]. We will design approximation algorithms for scheduling on unrelated machines, job shop scheduling, network embeddings and approximating metrics. We develop the packing formulation for these problems. Therefore they can be solved by the fast approximation algorithms for the packing problem in [18]. The key point is to construct the corresponding block solvers. We analyze the algorithms for block problems of these applications in detail, together with their running times. In this way we obtain approximation algorithms for these problems with improved bounds on running times.

1.1 Approximate Packing Problem

For a given accuracy tolerance $\varepsilon > 0$, the approximate packing problem is defined as:

$$(P_\varepsilon) \qquad \text{compute } x \in B \text{ such that } f(x) \leq (1+\varepsilon)\lambda^* \cdot \mathbb{1}.$$

According to the Lagrangian duality relation, $\lambda^* = \min_{x\in B} \max_{y\in P} y^T f(x) = \max_{y\in P} \min_{x\in B} y^T f(x)$, where $P = \{y \in \mathbb{R}^M | \sum_{m=1}^{M} y_m = 1, y_m \geq 0\}$. Denoting by $\Lambda(y) = \min_{x\in B} y^T f(x)$, an obvious observation is that $\Lambda(y) \leq \lambda^* \leq \lambda(x)$ for any pair x and y. Furthermore a pair $x \in B$ and $y \in P$ is optimal, if and only if $\lambda(x) = \Lambda(y)$. The corresponding approximate dual problem has the form:

$$(D_\varepsilon) \qquad \text{compute } y \in P \text{ such that } \Lambda(y) \geq (1-\varepsilon)\lambda^*.$$

The Lagrangian or price-directive decomposition method is usually applied in the algorithms, which is an iterative strategy to solve (P_ε) and (D_ε) by computing a sequence of pairs x and y, which approximate the optimal solution from above and below respectively. Grigoriadis and Khachiyan [13] developed an approximation algorithm to solve both the primal problem (P_ε) and the dual problem (D_ε) in $O(M(\log M + \varepsilon^{-2} \log \varepsilon^{-1}))$ iterations. Each iteration calls a standard t-approximate block solver $ABS(y,t)$ once, which solves the block problem for a given tolerance $t = O(\varepsilon)$:

$$\text{compute } \hat{x} = \hat{x}(y) \in B \text{ such that } y^T f(\hat{x}) \leq (1+t)\ \min\{y^T f(z) | z \in B\}.$$

Villavicencio and Grigoriadis [31] introduced a modified logarithmic potential function to avoid the ternary search approach in [13] and to simplify the analysis. The number of iterations is also $O(M(\log M + \varepsilon^{-2} \log \varepsilon^{-1}))$. Furthermore, Jansen and Zhang [18] reduced the number of iterations for both (P_ε) and (D_ε) to $O(M(\log M + \varepsilon^{-2}))$.

However, generally the block problem may be hard to approximate [3,8,9,19]. Indeed the problem of approximating metrics discussed in Section 6 has such a property, where the approximation ratio depends on the input size. This means that the assumption to have a block solver with both accuracy $t = O(\varepsilon)$ and

approximation ratio 1 is too strict (e.g. no PTAS). Therefore Jansen and Zhang [18] considered the case that only a weak approximate block solver is available. A weak (t, c)-approximate block solver $ABS(y, t, c)$ is defined as:

$$\text{compute } \hat{x} = \hat{x}(y) \in B \text{ such that } y^T f(\hat{x}) \leq c(1+t)\min\{y^T f(z) | z \in B\},$$

where $c \geq 1$ is the approximation ratio. The main goal is to solve the following primal problem (using the weak block solver):

$$(P_{\varepsilon,c}) \qquad \text{compute } x \in B \text{ such that } f(x) \leq c(1+\varepsilon)\lambda^* \cdot \mathbb{1}.$$

The corresponding dual problem is:

$$(D_{\varepsilon,c}) \qquad \text{compute } y \in P \text{ such that } \Lambda(y) \geq (1-\varepsilon)\lambda^*/c.$$

In [18] they proposed an approximation algorithm that for any accuracy $\varepsilon > 0$ solves the problem $(P_{\varepsilon,c})$ in $O(M(\log M + \varepsilon^{-2}\log\varepsilon^{-1}))$ coordination steps. Each step requires a call to the weak block solver $ABS(y, O(\varepsilon), c)$ and an overhead of $O(M \log\log(M\varepsilon^{-1}))$ arithmetic operations.

Plotkin et al. [23] considered the feasibility variants of packing problem for the linear case $f(x) = Ax$ where A is the coefficient matrix with M rows. The problem is solved by Lagrangian decomposition using exponential potential reduction. The number of iterations in their algorithm is $O(\varepsilon^{-2}\rho\log(M\varepsilon^{-1}))$, where $\rho = \max_{1\leq m\leq M}\max_{x\in B} a_m^T x/b_m$ is the width of B and b is the right hand side vector. Garg and Könemann [11] proposed a $(1+\varepsilon)$-approximation algorithm for the linear packing problem with $c = 1$ within $O(M\varepsilon^{-2}\log M)$ iterations, which is independent of the width ρ. Young [34] studied also the linear case of the packing problem (allowing weak block solver). His algorithm uses $O(\rho'(\lambda^*)^{-1}\varepsilon^{-2}\log M)$ calls to the block solver, where ρ' is a parameter similar to ρ. Furthermore, Charikar et al. [8] generalized the result in [23] for the packing problem with also $O(\varepsilon^{-2}\rho\log(M\varepsilon^{-1}))$ iterations. In fact the algorithm in [18] is the first one with a bound on the running time independent of parameters ρ, λ^* and c in the general case.

In this paper we will mainly compare our results with those by the algorithms in [23] and show that our algorithms can improve the running times. We will also show some typical example to give intuitive knowledge of the improvement. Furthermore, an important fact is that in most real applications the factor ε^{-1} plays a key role in running times. However, this does not receive sufficient attention yet and in some works the error tolerance ε is just regarded as a constant. In that case the resulting bounds on running times imply some large factors, which can not really indicate the quality of the algorithms. Hence we keep all the terms involving ε^{-1} for comparison.

1.2 Scheduling on Unrelated Machines

Let $\mathcal{J} = \{J_1, \ldots, J_n\}$ and $\mathcal{M} = \{M_1, \ldots, M_m\}$ be sets of jobs and machines, and $p_{ij} \geq 0$ the processing time of job J_j on machine M_i for $i = 1, \ldots, m$ and

$j = 1, \ldots, n$. The goal of the problem of *scheduling on unrelated machines* is to find a schedule such that each job is processed on exactly one machine and the *makespan* (maximum completion time) $C_{\max}$ is minimized. We consider the non-preemptive model here.

In general the problem is strongly NP-hard and there is no ρ-approximation polynomial time algorithm for any $\rho < 3/2$ unless $\mathcal{P} = \mathcal{NP}$ [22]. On the other hand, Lenstra et al. showed a 2-approximation algorithm. Afterwards, Plotkin et al. [23] presented a fast $(2 + \varepsilon)$-approximation algorithm using their approximation algorithm for the packing problem. Furthermore Jansen [16] improved the running time.

1.3 Job Shop Scheduling

In the *job shop scheduling* problem m machines and n jobs are given. Every job j consists of a sequence of operations, each of which is to be processed on a specific machine for a specific amount of processing time, subject to the constraint that on each machine at most one job is scheduled at any time. Here, a job can have more than one operations on a given machine (otherwise it is called *acyclic*). The goal is to minimize the makespan $C_{\max}$. We also consider the non-preemptive model.

The job shop scheduling problem is strongly $\mathcal{NP}$-hard [10,20]. Besides, the problem is very intractable in practice even for small instances [2,6]. Much attention has been paid to the approximation algorithms due to its hardness. However, there is no ρ-approximation polynomial time algorithm for job shop scheduling problem with $\rho < 5/4$, unless $\mathcal{P} = \mathcal{NP}$ [32]. Shmoys et al. [27] developed the first randomized and deterministic polynomial time algorithms for the general case with polylogarithmic approximation ratio. Their deterministic approximation bound was slightly improved by Schmidt et al. [26]. Later Goldberg et al. [12] presented polynomial time algorithms with improved approximation ratio by a doubly logarithmic factor. The derandomization of it strongly depends on the technique in [1] for NC and an integer packing problem, which essentially can also be solved by the algorithm in [18].

1.4 Network Embeddings

Given two n-node bounded degree graphs $G = (V, E_G)$ and $H = (V, E_H)$, a $1-1$ *embedding* of H in G is defined by specifying a path in G from i to j for each edge $(i, j) \in E_H$. The *dilation* of the embedding is the maximum number of edges on one of the paths used in G, and the *congestion* is the maximum number of paths used that contain the same edge in E_G.

In [21], an algorithm to embed H in G was proposed. The dilation and congestion are both $O(\alpha^{-1} \log n)$, where $\alpha = \min\{\delta(S)/|S| \mid S \subseteq V, |S| \leq n/2\}$ is the flux of G and $\delta(S)$ is the number of edges in E_G leaving S.

1.5 Approximating Metrics

Given a finite metric space induced by an undirected (weighted) graph, we want to embed it in a simpler metric space such that the distances are approximately preserved. Bartal [4] proposed the concept *probabilistic approximation* of a metric space by a set of simpler metric spaces. Given a graph $G = (V, E)$ and two finite metric spaces M_1 and M_2 defined on V, M_1 *dominates* M_2 if $d_{M_1}(u, v) \geq d_{M_2}(u, v)$ for all $u, v \in V$, where $d_M(u, v)$ is the metric distance between vertices u and v in M. Suppose S is a set of metric spaces on V. Assuming that each metric space in S dominates M, S is defined to α-*probabilistically approximate* M, if there is a probability distribution μ over S such that the expected distance distortion between each vertex pair in M in a space chosen from S according to μ is bounded by α.

In [4] a polynomial time algorithm to $O(\log^2 n)$-probabilistically approximate any metric space on $|V| = n$ vertices by a class of tree metrics was addressed. Moreover in [5] the approximation ratio was improved to $O(\log n \log\log n)$. However, the numbers of the tree metric spaces are exponentially large in both algorithms. Charikar et al. [8] developed a polynomial time algorithm to construct a probability distribution on $O(n \log n)$ trees metrics for any given metric space induced by a (weighted) graph G on n vertices, such that the expected stretch of each edge is no more than $O(\log n \log\log n)$. To decide the probability distribution μ, a linear program with exponential number of variables has to be solved. And there is only a weak block solver available.

2 Approximation Algorithms for the Packing Problem

Jansen and Zhang proposed a fast approximation algorithm $\mathcal{L}$ for the general packing problem with a weak approximate block solver [18]. The scaling phase strategy and Lagrangian coordination method are applied. In each scaling phase the relative error tolerance is set. In one iteration first the price vector is computed according to the known iterate. Then based on the price vector, the block solver is called to deliver a block solution. Afterwards an appropriate linear combination of the old iterate and block solution is computed as the new iterate. The iteration stops when the new iterate satisfies certain stopping rule. After one scaling phase, the error tolerance is halved and a new scaling phase starts until the given accuracy requirement is fulfilled. Besides, if the approximation ratio of the block solver is not too large, a similar algorithm $\mathcal{L}'$ can be used with less running time. With the initial setting in [18], the following results hold:

Proposition 1. *For a given relative accuracy $\varepsilon \in (0, 1]$, algorithm $\mathcal{L}$ stops with a solution x that satisfies $\lambda(x) \leq c(1 + \varepsilon)\lambda^*$ and performs a total of $N = O(M(\log M + \varepsilon^{-2} \log \varepsilon^{-1}))$ coordination steps.*

Proposition 2. *In the case of $\log c = O(\varepsilon)$, algorithm $\mathcal{L}'$ delivers a pair x and y with $\lambda(x) \leq c(1 + \varepsilon)\lambda^*$ and $\Lambda(y) \geq (1 - \varepsilon)\lambda^*/c$ within a total of $N = O(M(\log M + \varepsilon^{-2}))$ coordination steps.*

3 Scheduling on Unrelated Machines

With the definitions in Subsection 1.2, the scheduling problem on unrelated machines can be formulated as the following integer linear program

$$\begin{aligned} &\text{Min } \lambda \\ &\text{s.t. } \sum_{j=1}^{n} p_{ij}x_{ij} \le \lambda, \quad i = 1,\ldots,m; \\ &\quad \sum_{i=1}^{m} x_{ij} = 1, \quad j = 1,\ldots,n; \\ &\quad x_{ij} \in \{0,1\}, \quad i = 1,\ldots,m \text{ and } j = 1,\ldots,n. \end{aligned} \tag{1}$$

Here λ is the makespan. A feasible solution $x_{ij} = 1$ means that job J_j is executed on machine M_i. The algorithm in [22] is as follows: Firstly we guess the schedule length λ. Then the linear program relaxation of (1) can be solved. At last, the fractional solution is rounded to an integer one. Given the guessed makespan λ, the algorithm either concludes that there is no feasible solution with length less than λ, or delivers a schedule with makespan 2λ (even if there is no schedule of makespan λ). The main bottleneck in the algorithm is solving the corresponding linear program relaxation.

In [23], a fast $(2+\varepsilon)$-approximation algorithm was proposed by solving the linear program relaxation with their approximation algorithm for packing problem with the running time $O(\varepsilon^{-2}m^2n\log^2 n\log(m\varepsilon^{-1}))$. Furthermore, based on the algorithm in [13], Jansen [16] improved the running time to $O(m^2(\log m+\varepsilon^{-2}\log\varepsilon^{-1})(n+\log\log(m\varepsilon^{-1}))\log(m\varepsilon^{-1}))$.

We here apply the algorithm $\mathcal{L}'$ in [18] to solve the linear program relaxation. In the block optimization, for a price vector $y = (y_1,\ldots,y_m)^T \in P$ we compute the minimum dual value:

$$\Lambda(y) = \min_x \sum_{i=1}^{m} y_i \sum_{j=1}^{n} p_{ij}x_{ij} = \sum_{j=1}^{n} \min_x \sum_{i=1}^{m} y_i p_{ij}x_{ij}. = \sum_{j=1}^{n} \min_i y_i p_{ij}$$

Due to the structure of $B = B^1 \times \ldots \times B^n$, we can minimize the dual value separately over each B^j, $j = 1,\ldots,n$. For each B^j we just exactly compute the minimum weighted processing time $y_i p_{ij}$ over machine i with $p_{ij} \le \lambda$ and set the corresponding $x_{ij} = 1$. Each block optimization step takes $O(mn)$ time. On the other hand, the number of iterations to find a solution x with length less than $(1+\varepsilon)\lambda$ is $O(m(\log m+\varepsilon^{-2}))$, if the given λ exists. The numerical cost in each iteration is at most $O(m\log\log(m\varepsilon^{-1}))$. Finally, the number of iterations of binary search to find the $(1+\varepsilon)$-approximation of the optimal makespan λ^* is $O(\log(m\varepsilon^{-1}))$. With the rounding technique in [22,16], we have the following theorem:

Theorem 1. *For any given $\varepsilon > 0$, there exists a $(2+\varepsilon)$-approximation algorithm for scheduling problem on unrelated machines with a running time $O(m^2(\log m+\varepsilon^{-2})(n+\log\log(m\varepsilon^{-1}))\log(m\varepsilon^{-1}))$.*

The result in [16] improved the running time by a factor of $\log^2 n/\log \varepsilon^{-1}$ compared with that in [23]. Here our resulting running time is further improved by a factor of $\log \varepsilon^{-1}$. We notice that in [30] an algorithm for block structured packing problem was addressed and it can be also applied here. The running time is then $O(mn(\log m + \varepsilon^{-2}) \log m \log(m\varepsilon^{-1}))$.

4 Job Shop Scheduling

Denote by $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$ the set of machines, $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ the set of jobs, and $\mathcal{O} = \{O_{ij} | i = 1, \ldots, \mu_j, j = 1, \ldots, n\}$ the set of operations, with κ_{ij} indexing the machine which must process operation O_{ij}. Here μ_j is the number of operations of Job J_j, and $\mu = \max_j \mu_j$. Operation O_{ij} is the ith operation of job J_j, which requires processing time p_{ij} on a given machine $M_k \in \mathcal{M}$, where $k = \kappa_{ij}$. Let $P_{\max} = \max_j \sum_i p_{ij}$ the *maximum job length* of the instance. Because each job must be processed, the optimal makespan $C^*_{\max}$ must be at least $P_{\max}$. Moreover, we can also define $\Pi_{\max} = \max_k \sum_{\kappa_{ij}=k} p_{ij}$ the *maximum machine load* of the instance. Because each machine must process all operations assigned to it, $C^*_{\max}$ must be at least $\Pi_{\max}$, too.

The algorithm in [27] works as follows. First one instance is reduced to a special case in $O(m^2\mu^2 n^2)$ time, where $n = O(m^2\mu^3)$, $\Pi_{\max} = O(n^2\mu^2)$, $P_{\max} = O(n\mu^2)$, and $p_{\max} = \max_{i,j} p_{ij} = O(n\mu)$. Then for each job the randomized algorithm uniformly and independently selects an initial delay in $\{1, 2, \ldots, \Pi_{\max}\}$. It can be proved that if each job is scheduled continuously following its first operation starting after the chosen delay, then with high probability there is at most an $O(\log(m\mu)/\log\log(m\mu))$ congestion on any machine at any time. At last a "flattening" technique is applied to ensure that at any time on one machine there is at most one operation being processed. Almost all steps are deterministic except for the selection of initial delay. The approach to construct the deterministic algorithm is to regard the initial delays as a vector and it becomes a vector selection problem. The integer linear program is as follows:

$$\begin{aligned} &\text{Min } \lambda \\ &\text{s.t. } \sum_{j=1}^{n} \sum_{k=1}^{\Pi_{\max}} x_{jk} V_{jk}(i) \le \lambda, \quad && i = 1, \ldots, lm; \\ &\quad \sum_{k=1}^{\Pi_{\max}} x_{jk} = 1, && j = 1, \ldots, n; \\ &\quad x_{jk} \in \{0, 1\}, && j = 1, \ldots, n \text{ and } k = 1, \ldots, \Pi_{\max}. \end{aligned} \tag{2}$$

Here $l = P_{\max} + \Pi_{\max}$, $V_{jk}(i)$ is the vector whose ith component corresponds to the jth machine at time i with k delay, and x_{jk} is the variable to indicate whether V_{jk} is selected. There are $n\Pi_{\max} = O(n^3\mu^2)$ variables and $lm = O(mn^2\mu^2)$ constraints. And the computational bottleneck is solving the linear program relaxation of (2).

Our method is using algorithm $\mathcal{L}'$ to solve the linear program relaxation and round the fractional solution to obtain an integer one. The variables

$x_{jk} \in B = B^1 \times \ldots \times B^n$, where each B^j is a $\Pi_{\max}$-dimensional simplex: $B^j = \{(x_{j1}, \ldots, x_{j\Pi_{\max}}) | \sum_{k=1}^{\Pi_{\max}} x_{jk} = 1, x_{jk} \geq 0, k = 1, \ldots, \Pi_{\max}\}$, and each node corresponds to a particular delay for job j. In the block optimization, given a price vector $y = (y_1, \ldots, y_{lm})^T \in P$, we can show that the dual value is:

$$\Lambda(y) = \min_x \sum_{i=1}^{lm} \sum_{j=1}^{n} \sum_{k=1}^{\Pi_{\max}} y_i x_{jk} V_{jk}(i) = \sum_{j=1}^{n} \min_k \sum_{i=1}^{lm} y_i V_{jk}(i).$$

The last equality holds with the similar argument to Section 3. And $\Lambda(y)$ can be computed exactly in $O(m^2 n^5 \mu^4)$ time. The numerical overhead is bounded by $O(mn^2\mu^2 \log\log(mn^2\mu^2\varepsilon^{-1}))$. With the fractional solution to (2), the rounding technique in [25] and [24] can be applied to obtain an integer solution with congestion bounded by $O(\log(m\mu))$, which results in a deterministic algorithm. So the following theorem holds:

Theorem 2. *Given any $\varepsilon > 0$, there exists a deterministic $c(1+\varepsilon)$-approximation algorithm for the job shop scheduling problem in at most $O(m^2n^4\mu^4(\log(mn^2\mu^2) + \varepsilon^{-2})(mn^3\mu^2 + \log\log(mn^2\mu^2\varepsilon^{-1})))$ time, where $c = O(\log^2(m\mu))$.*

If the algorithm in [23] is applied, the width ρ has a bound $O(n^2\mu^2)$. In addition, it is an algorithm for decision problem. Therefore a number $O(\log(n\mu^2\varepsilon^{-1}))$ of iterations of binary search are needed to get an approximate solution. The total running time is $O(mn^7\mu^6\varepsilon^{-2}\log(mn^2\mu^2\varepsilon^{-1})\log(n\mu^2\varepsilon^{-1}))$. The algorithm by Vaidya [28] can also be used to solve the linear program here and it has a running time of $\Omega(n^{10.5}\mu^7\log(m\mu))$. Since in our result the running time is dominated by $O(m^3n^7\mu^6\varepsilon^{-2})$, the improvement is $\log(mn^2\mu^2\varepsilon^{-1})\log(n\mu^2\varepsilon^{-1})/m^2$. In a typical example in [6], $m = n = \mu = 10$ and if ε is 0.01, the improvement is a factor $0.35\log^2 100$. In fact in the real cases n and μ will be much larger than m and our improvement is also larger.

5 Network Embeddings

The bottleneck of the algorithm in [21] is also to find an approximate solution to a packing problem, which corresponds to the problem of routing the edges in H. Suppose for each path p in G corresponding to an edge $e_H \in E_H$, there is an indicator variable $x_{e_H}(p)$. The integer linear program is as follows:

$$\begin{aligned} &\text{Min } \lambda \\ &\text{s.t.} \sum_{e_G \in p} x_{e_H}(p) \leq \lambda, \qquad \text{for all } e_G \in E_G; \\ &\quad \sum_{p \in P_{e_H}} x_{e_H}(p) = 1, \qquad \text{for all } e_H \in E_H; \\ &\quad x_{e_H}(p) \in \{0,1\}, \qquad \text{for all paths } p \in P_{e_H} \text{ and all } e_H \in E_H. \end{aligned} \tag{3}$$

Here $P_{e_H} = \{$*paths in G corresponding to $e_H \in H$ with lengths no more than λ*$\}$. In addition, the set $B = B^1 \times \ldots \times B^{|E_H|}$ is a product of simplices with various

dimensions (B^{e_H} is a $|P_{e_H}|$-dimensional simplex), each of which guarantees that one path in G is selected for an edge $e_H \in E_H$. (3) ensures that the dilation and congestion are all bounded by λ.

We apply algorithm $\mathcal{L}'$ to solve (3). There are $O(n)$ constraints for G is a bounded degree graph. Therefore the number of iteration is $O(n(\log n + \varepsilon^{-2}))$. Given a price vector y, the minimum dual value is

$$\Lambda(y) = \min_p \sum_{e_G \in p} y_{e_G} x_{e_H}(p) = \sum_{e_G \in p} \min_p y_{e_G} x_{e_H}(p),$$

which means finding a minimum-cost path in G corresponding to an edge $e_H \in E_H$, with the length at most λ. By dynamic programming, the minimum-cost path can be found in $O(n\lambda)$ time. After solving the linear program relaxation of (3), the rounding technique in [24,25] can be used to obtain an integer solution. Therefore the following theorem holds:

Theorem 3. *For any given $\varepsilon > 0$, a graph $H = (V, E_H)$ can be embedded in another graph $G = (V, E_G)$ with $|V| = n$ in $O(n^2(\log n + \varepsilon^{-2})(\alpha^{-1} \log n + \log\log(n\varepsilon^{-1})))$ time with both the dilation and the congestion bounded by $c(1+\varepsilon)$, where $c = O(\alpha^{-1} \log n)$.*

If the algorithm for packing problem in [23] is applied, the running time is $O(n^3\varepsilon^{-2}\alpha^{-1} \log n \log(n\varepsilon^{-1}) \log(\varepsilon^{-1}\alpha^{-1} \log n))$. In addition, the running time of the algorithm in [29] is $\Omega(n^2 \log n(n^2\alpha^{-1} \log n + \mathcal{M}(n)))$, where $\mathcal{M}(n)$ is the time needed to invert an $n \times n$ matrix. Here compared with [23] our improvement is a factor of $n \log(n\varepsilon^{-1}) \log(\varepsilon^{-1}\alpha^{-1} \log n)$.

6 Approximating Metrics

We consider the algorithm in [8]. The idea of establishing the probability distribution is as follows. Let M be a finite metric space induced by a (weighted) undirected graph $G = (V, E)$ with $|V| = n$, and $S = \{T_1, T_2, ..., T_N\}$ be a set of tree metrics on V. Here N could be exponentially large and we will find a polynomial-size subset among the N tree metrics. Assume that each of the metrics in S dominates G, i.e., $d_{T_i}(u, v) \geq d_G(u, v)$ for every pair of vertices (u, v) and every $i \in \{1, \ldots, N\}$. Let $c(e)$ be the length of an edge $e \in E$. Suppose that $c(u, v) \leq d_G(u, v)$. Using a real value $0 \leq x_i \leq 1$ for each $T_i \in S$, which satisfies $\sum_{i=1}^{N} x_i = 1$, to represent a probability distribution on S, the probabilistic approximation of G by S can be obtained by the following linear program formulation:

$$\begin{aligned} &\text{Min } \lambda \\ &\text{s.t. } \sum_{i=1}^{N} d_{T_i}(e) x_i \leq \lambda c(e), \quad \text{for all } e \in E; \\ &\quad \sum_{i=1}^{N} x_i = 1; \\ &\quad x_i \geq 0, \qquad\qquad i = 1, \ldots, N. \end{aligned} \tag{4}$$

There are m packing constraints and the set $B = \{(x_1, \ldots, x_N) | \sum_{i=1}^{N} x_i = 1, x_i \geq 0, 1 \leq i \leq N\}$ is a simplex. The block optimization is to find the minimum dual value for a pre-computed price vector $y = (y_1, \ldots, y_m)^T$:

$$\Lambda(y) = \min_x \sum_{e \in E} \sum_{i=1}^{N} y_e \frac{d_{T_i}(e)}{c(e)} x_i = \min_x \sum_{i=1}^{N} x_i \sum_{e \in E} y_e \frac{d_{T_i}(e)}{c(e)} = \min_i \sum_{e \in E} y_e \frac{d_{T_i}(e)}{c(e)}.$$

Regarding the components of modified price vector $y_e/c(e)$ for all $e \in E$ as a weight function assigned to edges, the goal is to find a tree such that the average (y/c)-weighted edge lengths of G are minimized. The block problem is in fact the minimum communication cost spanning tree (MCCST) problem on metric spaces, which is however $\mathcal{NP}$-hard [10,15,33]. The first deterministic polynomial time approximation algorithm was addressed in [5] and [7] independently. And the approximation ratio is $O(\log n \log\log n)$. The worst case stretch of any edge is bounded by $O(n)$.

It is worth noting that in this problem, only weak approximate block solver is available, similar to the multicast congestion problem in communication networks studied in [3,19]. In [8] the algorithm proposed in [23] was generalized to the case of large approximation ratio. Therefore they developed an algorithm for approximating metrics and the running time is $O(n\varepsilon^{-2} \log(m\varepsilon^{-1}) \log(n\varepsilon^{-1})\beta)$, given a relative error tolerance $\varepsilon > 0$ and with the technique of reducing the width, where β is the time to solve the minimum communication cost spanning tree problem approximately.

The algorithm $\mathcal{L}$ can be applied for the case of large approximation ratio, without increasing the number of iterations much. Hence we have the following theorem:

Theorem 4. *For a given $\varepsilon > 0$, any finite metric space induced by a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$ can be $c(1+\varepsilon)$-probabilistically approximated by a probability distribution on $O(n \log n)$ tree metrics in $O(m(\log m + \varepsilon^{-2} \log \varepsilon^{-1})(\beta + m \log\log(m\varepsilon^{-1})))$ time, given a c-approximate solver of the MCCST problem, where $c = O(\log n \log\log n)$ and β is the running time of the MCCST solver.*

The dominant term of the running time of our algorithm is $O(m\varepsilon^{-2}\beta \log \varepsilon^{-1})$. Compared with that in [8], in the case of sparse graph where $m = O(n)$, our improvement is $\log^2(n\varepsilon^{-1})/\log \varepsilon^{-1}$.

7 Concluding Remarks

In this paper we have presented approximation algorithms for scheduling on unrelated machines, job shop scheduling, network embeddings and approximating metrics as applications of the fast approximation algorithms for the packing problem in [18]. The running times are improved for above problems. An interesting open problem is whether we can apply the technique for the packing problems with block structures in [30] to the algorithms in [18] for the case of only a weak block solver.

Acknowledgment. The author thanks Klaus Jansen and Yan Dong for helpful discussion and comments.

References

1. N. Alon and A. Srinivasan, Improved parallel approximation of a class of integer programming problems, *Proceedings of the 23rd International Colloquium on Automata, Languages and Programming*, ICALP 1996, 562–573.
2. D. Applegate and W. Cook, A computational study of the job-shop scheduling problem, *ORSA Journal of Computing*, 3 (1991), 149–156.
3. A. Baltz and A. Srivastav, Fast Approximation of Minimum Multicast Congestion – Implementation versus Theory, *Proceedings of the 5th Conference on Algorithms and Complexity*, CIAC 2003.
4. Y. Bartal, Probabilistic approximation of metric spaces and its algorithmic applications, *Proceedings of the 37th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 1996, 184–193.
5. Y. Bartal, On approximating arbitrary metrics by tree metrics, *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, STOC 1998.
6. J. Carlier and E. Pinson, An algorithm for solving the job-shop problem, *Management Science*, 35 (1989) 164–176.
7. M. Charikar, C. Chekuri, A. Goel and S. Guha, Rounding via trees: deterministic approximation algorithms for group steiner trees and k-median, *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, STOC 1998.
8. M. Charikar, C. Chekuri, A. Goel, S. Guha and S. Plotkin, Approximating a finite metric by a small number of tree metrics, *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 1998, 379–388.
9. G. Even, J. S. Naor, S. Rao and B. Schieber, Fast approximate graph partitioning algorithms, *SIAM. Journal on Computing*, 6 (1999), 2187–2214.
10. M. Garey and D. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, NY, 1979.
11. N. Garg and J. Könemann, Fast and simpler algorithms for multicommodity flow and other fractional packing problems, *Proceedings of the 39th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 1998, 300–309.
12. L. A. Goldberg, M. Paterson, A. Srinivasan and E. Sweedyk, Better approximation guarantees for job-shop scheduling, *SIAM Journal on Discrete Mathematics*, 14 (2001), 67–92.
13. M. D. Grigoriadis and L. G. Khachiyan, Coordination complexity of parallel price-directive decomposition, *Mathematics of Operations Research*, 2 (1996), 321–340.
14. M. D. Grigoriadis and L. G. Khachiyan, Approximate minimum-cost multicommodity flows in $O(\varepsilon^{-2}knm)$ time, *Mathematical Programming*, 75 (1996), 477–482.
15. T. C. Hu, Optimum communication spanning trees, *SIAM Journal on Computing*, 3 (1974), 188–195.
16. K. Jansen, Approximation algorithms for fractional covering and packing problems, and applications, Manuscript, (2003).
17. K. Jansen and R. Solis-Oba, An asymptotic fully polynomial time approximation scheme for bin covering, *Proceedings of 13th International Symposium on Algorithms and Computation*, ISAAC 2002.
18. K. Jansen and H. Zhang, Approximation algorithms for general packing problems with modified logarithmic potential function, *Proceedings of 2nd IFIP International Conference on Theoretical Computer Science*, TCS 2002.

19. K. Jansen and H. Zhang, An approximation algorithm for the multicast congestion problem via minimum Steiner trees, *Proceedings of 3rd International Workshop on Approximation and Randomized Algorithms in Communication Networks*, ARANCE 2002.
20. E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan and D. B. Shmoys, Sequencing and scheduling: algorithms and complexity, in *S. C. Graves et al. (eds), Handbooks in Operations Research and Management Science, Vol 4: Logistics of Production and Inventory*, Elsevier, (1993) 445–522.
21. T. Leighton and S. Rao, An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms, *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, FOCS 1988, 422–431.
22. J. K. Lenstra, D. B. Shmoys and E. Tardos, Approximation algorithms for scheduling unrelated parallel machines, *Mathematical Programming*, 24 (1990), 259–272.
23. S. A. Plotkin, D. B. Shmoys and E. Tardos, Fast Approximation algorithms for fractional packing and covering problems, *Mathematics of Operations Research*, 2 (1995), 257–301.
24. P. Raghavan, Probabilistic construction of deterministic algorithms: Approximating packing integer programs, *Journal of Computer and System Science*, 37 (1988), 130–143.
25. P. Raghavan and C. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs, *Combinatorica*, 7 (1987), 365–374.
26. J. P. Schmidt, A. Siegel and A. Srinivasan, Chernoff-Hoeffding bounds for applications with limited independence, *SIAM Journal on Discrete Mathematics*, 8 (1995), 223–250.
27. D. B. Shmoys, C. Stein and J. Wein, Improved approximation algorithms for shop scheduling problems, *SIAM Journal on Computing*, 23 (1994), 617–632.
28. P. M. Vaidya, Speeding up linear programming using fast matrix multiplication, *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 1989, 332–337.
29. P. M. Vaidya, A new algorithm for minimizing convex functions over convex sets, *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 1989, 338–343.
30. J. Villavicencio and M. D. Grigoriadis, Approximate structured optimization by cyclic block-coordinate descent, *in H. Fischer et al. (Eds.), Applied Mathematics and Parallel Computing – Festschrift for Klaus Ritter*, Physica-Verlag, Heidelberg (1996), 359–371.
31. J. Villavicencio and M. D. Grigoriadis, Approximate Lagrangian decomposition with a modified Karmarkar logarithmic potential, *Network Optimization, P. Pardalos, D. W. Hearn and W. W. Hager, Eds, Lecture Notes in Economics and Mathematical Systems 450, Springer-Verlag, Berlin*, (1997), 471–485.
32. D. P. Williamson, L. A. Hall, J. A. Hoogeveen, C. A. Hurkens, J. K. Lenstra, S. V. Sevast'yanov and D. B. Shmoys, Short shop schedules, *Operations Research*, 45 (1997), 288–294.
33. B. Y. Wu, G. Lancia, V. Bafna, K. Chao, R. Ravi and C. Y. Tang, A polynomial time approximation scheme for minimum routing cost spanning trees, *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 1998.
34. N. E. Young, Randomized rounding without solving the linear program, *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms*, SODA 1995, 170–178.

Design Patterns in Scientific Software

Henry Gardner

Computer Science, FEIT, Australian National University, Canberra, ACT 0200, Australia
Henry.Gardner@anu.edu.au

Abstract. This paper proposes that object-oriented design patterns can greatly help with the design and construction of scientific software. It describes a method of teaching design patterns which introduces patterns as they are used in refactoring, extending and reusing a computational science case study. The method has been taught into a graduate level eScience curriculum for three years.

1 Introduction: Teaching Computational Science and eScience

Many universities are now offering courses and degree programs in computational science. Although individual approaches differ, these programs usually have a core component of numerical analysis combined with the use of computers to solve real-world problems in a particular application domain. Courses are usually taught by departments of Mathematics and Computer Science or in the application disciplines.

Starting in the early 1990s, several computational science courses were introduced into the curriculum of the Australian National University (ANU). Eventually the cross-disciplinary interest in computational science resulted in the establishment of one new undergraduate and two new graduate programs. The undergraduate Bachelor of Computational Science (BComptlSci)[1] has a number of offerings by the departments of Mathematics and Computer Science. The new graduate programs, in "eScience"[2], are offered by the Department of Computer Science as conversion degrees which take students with previous qualifications in science and engineering and provide them with a range of computing skills which will not only help them "convert" to being computational scientists in their chosen disciplines but also prepare them for a career in the mainstream Information Technology industry. The eScience programs attempt to provide students with a rigorous introduction to a modern programming language, software engineering, computer graphics, High-Performance Computing (HPC) optimisation, networking and human-computer interaction.

Much of the eScience syllabus relates to programming and how to program better. Experience in presenting this material has led the author to propose that the adoption of one aspect of modern programming practice could significantly improve the productivity of mainstream computational scientists. This topic is object-oriented (OO) design patterns. The motivation behind this proposition

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 776–785, 2004.
© Springer-Verlag Berlin Heidelberg 2004

will be explained in Sections 2 and 3 below. An approach to teaching patterns using a computational science case study will be explained in Section 4. Further details, together with an example, are presented in Sections 5 and 6 and the paper is concluded with Section 7.

2 Programming in the Small, in the Large, and in Between

At an elementary level, computer science departments offer courses which are concerned with "programming in the small". Students learn the syntax and semantics of a particular programming language as well as basic algorithms and data structures and they design and code small, well-contained, programs. Some computer science departments teach first-year courses in a procedural language such as C (hardly ever in Fortran). For pedagogical reasons, others prefer to use more exotic "functional languages" such as Haskell. Many institutions, such as ours, offer object-oriented (OO) programming languages from the very first course (many use Java™; at ANU we have used Eiffel). Whether they start out with OO programming or not, computer science students are usually deeply immersed in the OO paradigm by the end of their second year. Object-oriented programming languages are also dominant in industry.

"Programming in the large" is usually called "software engineering". It has a heavy emphasis on the management of software projects as well as on the design of large software systems for reliable use and reuse and for easy maintenance. These days the dominant methodologies for software engineering are also object-oriented in nature.

Somewhere in between, is the design and construction of medium-sized "subsystems" of software. This subsystem level is the main focus of the idea of using recurring solutions to design problems (or "patterns") to help design good software. In computing, the subject of design patterns in OO software is now very influential. It is a sign of the maturity, but also of the difficulty, of the discipline writing good, object-oriented software.

Following a series of discussions and technical meetings in the early 1990s, a group of computer scientists published a book "Design Patterns - Elements of Reusable Software"[3] which has become regarded as the seminal work on design patterns. It is often known as the "Gang of Four" – GoF – book. This work describes 23 patterns which are grouped into three categories: "creational", "structural" and "behavioral".

In spite of its popularity, the GoF book is not an easy read! The authors have tried to be very precise about the patterns that they describe. For each of the patterns, they consider a number of motivations and possible settings where the pattern might occur. They provide Unified Modelling Language (UML) diagrams for the pattern and then describe its implementation, advantages and disadvantages. There is a large amount of detail and it is sometimes difficult to sort out the wood from the trees. The main languages being considered are C++ and Smalltalk.

3 Can Computational Scientists Be Taught to Program Better?

There is considerable evidence, anecdotal and otherwise, that scientific software is expensive to maintain and reuse. Paul Dubois, in his book "Object Technology for Scientific Computing" notes that scientific software can often have a poor quality of life, can find it difficult to reproduce, can have a high cost of living and "most scientific programs will die an early death"[4]. The "poor quality of life" idea means that coding errors may lie uncaught for some time and might even be interpreted as limitations in the science being modelled. One study which exposes these problems in a systematic way is the work by Les Hatton[5] where he compares several large, commercial software packages for the interpretation of seismic data. Because of coding errors, the difference in interpretation between several of the packages meant that they would have given completely different qualitative interpretations of the sample dataset.

So there does seem to be ground for concluding that computational scientists have something to learn about programming. If computer scientists can teach them, then it will probably be within the OO paradigm. But the principles of OO programming "are subtle and require extensive training and practice to get right" [6]. They are also, sometimes, at variance with the necessity to optimise computational programs for speed on HPC architectures (more objects can mean worse performance).

It would seem that computational scientists need to access a middle ground in their study of OO design and programming. They could benefit from a study of the basic programming principles and language elements but, in order to make practical progress, they need to have access to "recipe like" patterns of OO subsystems which are appropriate for scientific software.

4 A New Approach to Teaching Design Patterns

These days many introductory programming books make some reference to design patterns. There are also specialist design patterns books which attempt to explain the classic GoF book using different examples and different implementation languages (for example, [7,8]). These references tend to introduce patterns in one of two ways: The first way is to follow the basic structure of the GoF book – starting with creational patterns and then moving on to consider structural and behavioral patterns or possibly switching order but keeping the groupings the same. They often translate the GoF ideas into a different programming language and use different examples. The second way that patterns are often explained, is to find patterns in software that students would be familiar with (typically in standard packages from the Java programming environment) and to explain how and why those patterns came to be there.

As part of the eScience program at ANU, the students study design patterns following a syllabus which does away with much of the detail of the GoF book. They end up only looking at some 13 out of the classic 23 patterns and they are presented with one main motivation for patterns: patterns can be used to

keep subsystems well encapsulated. This good encapsulation means that subsystems are better able to be extended and reused than they would otherwise have been. Patterns also help to implement the reuse and adaptation of subsystems. The students go about building a software application which is a simpler version of a case study from computational science. They proceed, at first, in a common-sense manner. Once the software is reasonably complicated, the students then re-engineer (or "refactor") it using design patterns. The application is then extended to enhance its functionality and to demonstrate the flexibility of the software architecture.

This treatment of design patterns is influenced by a particular methodology of software development known as "Executable UML" [9] and it complements a software engineering course which is taken by the eScience students at ANU. Executable UML emphasises the analysis of a complicated software system by splitting it up into well-defined "domains" and "sub-domains". It also advocates the use of a consistent subset of UML. But Executable UML does not deal with design patterns or with the programming level of software at all. Instead, it proposes that software systems can be constructed entirely at a the level of UML modelling and then be translated into a target programming language using a "model compiler". This level of abstraction would appear to be way off in the future for practical computational scientists.

5 The Case Study: A Data-Viewer for Fusion Plasma Physics

Magnetic fusion experiments use large toroidal (doughnut-shaped) magnetic fields to confine an ionised gas (or plasma) at temperatures approaching that of the sun. The largest in the world, the JET tokamak in the UK, has a major radius of 2.96m and an (interior) height of 4.2m. If the conditions are right, and the appropriate fuels are being used, nuclear fusion reactions occur and release energetic particles and radiation. The eventual goal is to convert the released energy to heat and to use it to generate electricity. This should harness a new form of nuclear power which will have significant safety and environmental advantages over present, nuclear-fission power plants.

There are many magnetic fusion experiments around the world including a national facility in Australia. They operate as a series of pulsed "shots" where a plasma discharge is fired-up and confined over some tenths of a second. During this shot a huge amount of data is collected by diagnostic equipment and is stored as wave-form-like tables of signal-amplitude versus time.

MDSplus[10] is a data acquisition and analysis system which has been developed especially for magnetic fusion experiments. It is widely used internationally, on experiments worth many hundreds of millions of dollars. It was developed jointly by

- MIT Plasma Science and Fusion Center, Cambridge, MA, USA
- Los Alamos National Laboratory, Los Alamos, NM, USA
- Istituto Gas Ionizzati, Padova, Italy

Versions of MDSplus are Globus[11] enabled.

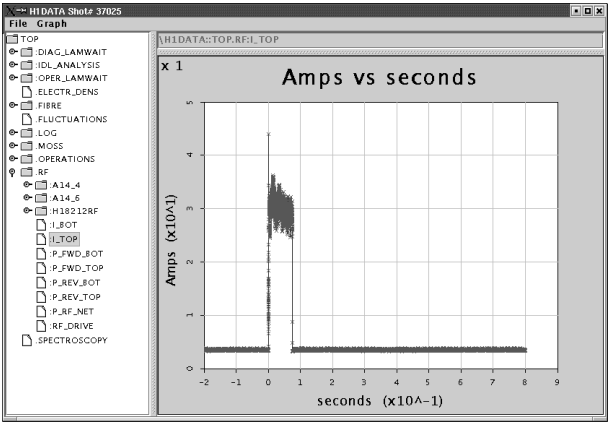


Fig. 1. View of the main window of the case study showing an experiment tree and some waveform data.

One of the tools supplied with MDSplus is a program known as "Scope" (from "oscilloscope"). The tool provides a set of X-Y plots of signal traces versus time. "jScope"[12] is a Java version of Scope which promises greater portability and enhanced networking facilities. The evolution of fusion research into a small number of large experiments, each supporting an international research base of scientists, has made remote collaboration essential. Using a tool which is able to display (near real-time) experimental data at a remote site should enable collaborating scientists to participate in an experimental program as if they were actually in the machine control room.

Because of the local interest in fusion science at ANU, and because the jScope data-viewer combined many aspects of programming that students would be interested in (the internet, a graphical user interface, visualisation of scientific data and dealing with a database of scientific data) the author decided to use something like it for teaching advanced programming and design patterns. But, because jScope is some tens of thousands of lines of code, it was deemed preferable to construct a much simpler version of it from scratch for the eScience course. The eScience students construct this simpler program, which is called "EScope", themselves.

Students start by being given classes which enable basic networked access to a sample MDSplus database of fusion diagnostic data. They have to make these work and then build up a Java-Swing^{TM} graphical user interface (GUI) to enable a user to connect to the MDSplus database over the Internet and to download a tree-like representation of all of the files in that database and to plot up selected waveform files. They go about these initial exercises using "good programming principles" as taught in introductory textbooks. Eventually they construct a program which looks something like the one in Fig.1.

The interesting work with patterns starts after the students have constructed an initial version of the program (without the actual graphics). They are encouraged to consider the "domains of concern" of their program. It is clear that these fall into two very distinct categories:

1. Interactions with the MDSplus data server. Classes involved in these activities make the network connections and send and receive information to and from the server using a special command language. (Data is received in special "packets" which must be decoded to be used by the rest of the program.)
2. The appearance and function of the main GUI window and associated dialog windows. By selecting appropriate menu options, a user is able to ask that a network connection be made and to select a particular data tree. After the tree is displayed graphically, selecting a particular node of the tree using the mouse will cause information about that data file to be displayed.

Both of these domains involve quite different subject matter and can be discussed using different specialised language. If they were to be designed to be independent of each other, then there would be a greater chance that each would be able to be reused (or partially reused) by another program with similar requirements *on that particular subject matter* in the future. This is the starting point for the treatment of the Facade and Mediator patterns.

In order to properly deal with OO class structures, it is useful to have a graphical notation.

5.1 A Simplified Version of UML

Prior to the Unified Modelling Language, there were several schools of thought about modelling OO software. UML brought the notation of some of them together. Although "Unified" carries with it ideas of consistency, it turns out to be quite difficult to sort through the diagramatic conventions used as "UML" by different authors. Because of this, this author has decided to be unafraid of introducing his own "simplified UML subsets" ("sUMLs") in his teaching. These sUMLs try to be very close to "standard" UML but are more consistent and satisfactory to use for first-time students. The figures in this paper use one sUMLs to describe *Class Diagrams* which are collections of rectangular boxes with lines drawn between them. The boxes are meant to represent classes and interfaces. The lines are meant to represent the *relationships* of association, dependency and specialisation (inheritance and implementation). The idea is that the diagrams should display the connectivity between components of a software system in much the same way that a circuit diagram displays the connectivity between electronic components.

A schematic, sUMLs, class diagram for the simple EScope system without graphics is shown in Fig. 2.

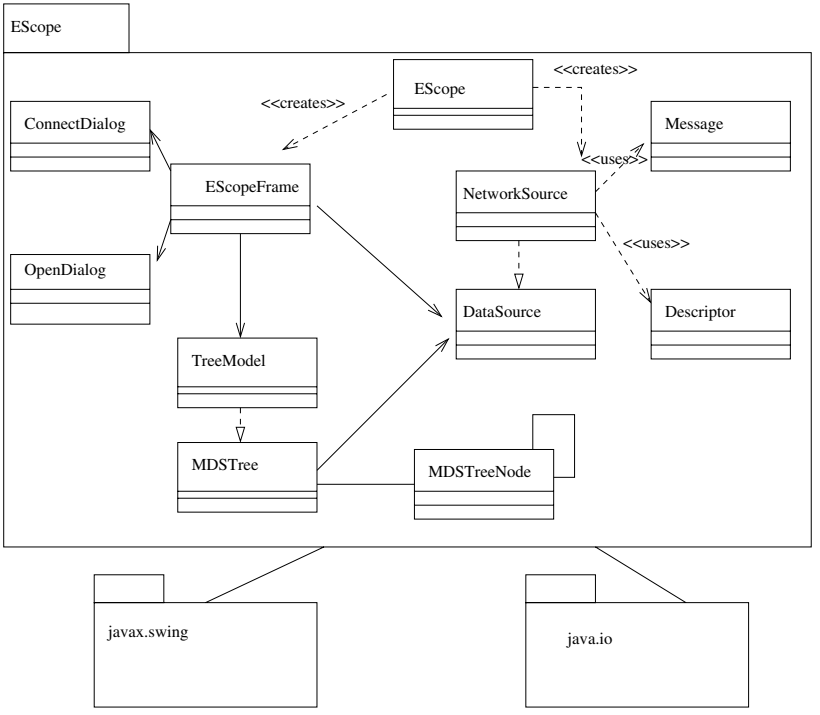


Fig. 2. A schematic UML-like class diagram for the simple EScope system before restructuring.

6 Introducing Patterns

The Facade pattern wraps up a subsystem and regulates the flow of control and data into and out of that subsystem. The Mediator pattern regulates the flow of control and data within a subsystem. The students learn a particular implementation of both patterns which uses interfaces to the facade classes to increase the decoupling between the different domains. In our Java implementation of the program, the subsystems are located in different directories. The main program, in the top directory, is responsible for constructing each subsystem by constructing their facade classes. It then initialises the subsystems by passing them references to each other. A class diagram for the re-engineered software using the Facade and Mediator patterns is shown in Fig. 3.

Further patterns are now encountered by modifying and extending the software to incorporate a graphics panel for the waveform data, to implement a local caching of viewed data, and to make other changes as mentioned below.

The Adapter pattern adapts a class of one interface to satisfy the requirements of another. One way of achieving this is to have an associative link from the adapting class, which implements the new interface, to the other which implements the old interface. This implementation (the “Object Adapter”) also

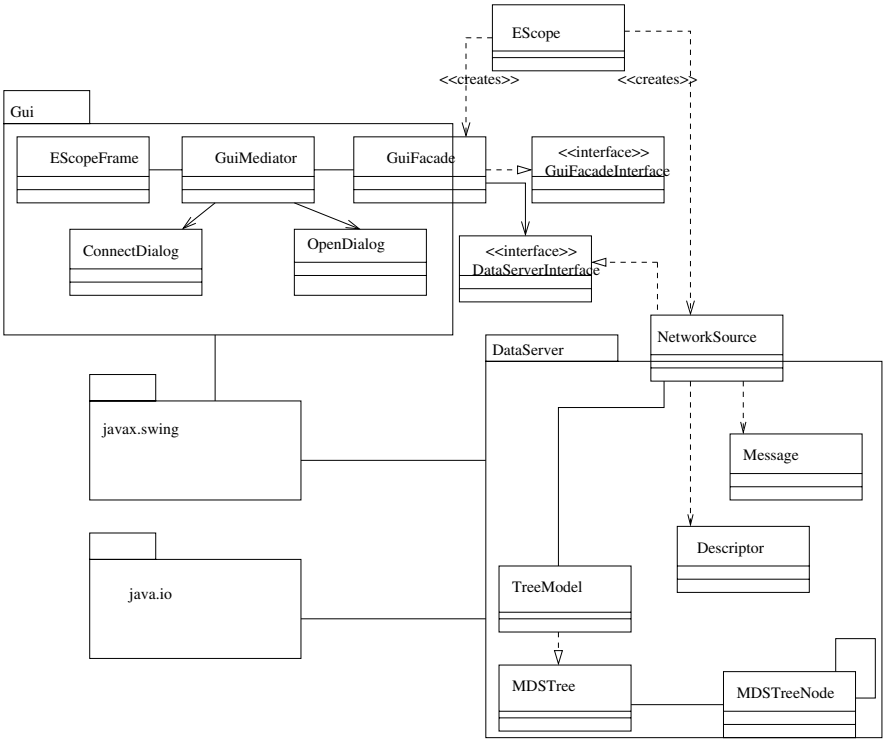


Fig. 3. A schematic, UML-like, class diagram for the simple EScope system after restructuring.

forms the basis of the Proxy pattern. In our case study, the Adapter is used to change the interface to the DataServer domain without modifying the, legacy, NetworkSource class. The Proxy is used to cache graph data after extending the software to incorporate a Graphics domain.

The tree data-structure used to store data retrieved from the server can be implemented using the, structural, Composite pattern. The behavioral, Iterator pattern can then be used to enumerate the tree nodes in a way which is independent of the structure itself. This pattern gives an important decoupling between access and structure of data and is also used extensively in the Java collections framework.

Central to early ideas of OO programming is the idea of simulation of a state machine. A state machine is an "active object" (or an active collection of objects) whose behavior changes depending on different modes. In a straightforward implementation, these modes and behaviors can be maintained by flags and nested decision statements. There is a special State pattern which does away with the complicated logic which results from these decision statements and which makes it easier to modify and extend state behavior. This is a more complicated pattern; we teach it with reference to the the facade of the data server domain.

The Observer pattern can be illustrated with reference to the GUI domain of our system. Creational patterns can be used in the initialisation phase of the EScope system. These can be basic Factory patterns or a Builder pattern depending on the complexity of the initialisation task. At the level of the main program, the Singleton pattern is useful to restrict the number of subsystem objects to being one of each type. The course quickly covers up to 5 of the Factory patterns which brings the total to 13 – more or less depending on the enthusiasm of the lecturer and students.

In terms of performance improvement for effort, the most important patterns in this case study are the Facade, Mediator, Adapter, Proxy, Observer and State.

7 Conclusion

This paper has described a new approach to teaching object-oriented design patterns which uses the refactoring, reuse and extending of a case study from fusion science to introduce some 13 of the 23 classic design patterns[3]. The case study involves exciting "big science" which is appealing to students with a science and engineering background. It incorporates many aspects of modern programming such as the internet, graphics, graphical user interfaces and a database. It is also complex enough to motivate the study of design patterns. Several important design patterns are clearly illustrated by the application and their uses include the encapsulation of subsystems, the adaptation of interfaces and the representation of the state behavior of active objects. Students are also introduced to 2D graphics and they build a data-viewer which could, in the future, be adapted to a range of other applications. The course is now in its fourth year and recorded student satisfaction has been comparatively high. Feedback from students, and from their performance in practical and written examinations, has been fed into annual revisions of the course. Over the years, the course content has evolved from having a large emphasis on practical programming and only a brief mention of patterns, to having a large emphasis on patterns incorporating many other examples outside of the case study, to the format described here which emphasises several key patterns which are strongly linked to the case study. As part of the eScience program, students are required to complete a programming project. Efforts are made to get students to use a similar pattern framework in these projects and some of them do this quite well.

Several factors will continue to push the scientific software community in the direction of object-orientation. These include new generations of students who have been trained in languages such as Java, the evolution of Fortran to incorporate OO features[13] and the optimisation of OO compilers and virtual machines to have higher, scientific-level performance. The author is not of the opinion that computational scientists should rush out and learn all there is to know about OO programming because there is a considerable overhead in doing so. On the other hand, the central idea of the encapsulation and reuse of subsystems, embodied in some of the design patterns discussed here, should be able to be appreciated, and even implemented, by computational scientists using, for example, Fortran 95.

Acknowledgments. The author wishes to thank many people for their help and encouragement: Shayne Flint and Clive Boughton taught him about Executable UML and Shayne explained how it might be translated into Java. Boyd Blackwell, Rhys Hawkins and Rod Harris contributed to the development of the EScope case study. Gabriele Manduchi made jScope available and Tom Fredian has been the main contact for MDSplus. Special thanks to Boyd Blackwell who has managed the MDSplus installation at the H-1NF experiment at ANU and who has supplied the inspiration and the sample data for the case study.

References

1. Australian National University: ANU Bachelor of Computational Science website. (2004) `http://wwwmaths.anu.edu.au/study/bcomptlsci/`. Last accessed 29 January 2004.
2. Australian National University: ANU eScience website. (2004) `http://eScience.anu.edu.au`. Last accessed 29 January 2004.
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object Oriented Software. Addison-Wesley (1995) ISBN 0201633612.
4. Dubois, P.F.: Object Technology for Scientific Computing: Object-Oriented Numerical Software in Eiffel and C. Prentice Hall PTR (1997) ISBN 0-13-518861-X.
5. Hatton, L.: The t experiments: Errors in scientific software. IEEE Computational Science and Engineering **4** (1997) 27–38
6. Wiener, R.: Watch your language! IEEE Software **15** (1998) 55–56
7. Cooper, J.W.: Java Design Patterns: A Tutorial. Addison Wesley (2000) ISBN 0-201-48539-7.
8. Stelting, S., Maassen, O.: Applied Java Patterns. Sun Microsystems Press (2002) ISBN 0-13-093538-7.
9. Mellor, S., Balcer, M.: Executable UML, A foundation for Model-Driven Architecture. Addison-Wesley, Indianapolis, IN (2002)
10. MDSplus contributors: MDSplus Web site. (2004) `http://www.mdsplus.org/intro/`. Last accessed 29 January 2004.
11. Globus Alliance: Globus website. (2004) `http://www.globus.org/`. Last accessed 29 January 2004.
12. MDSplus contributors: jScope website. (2004) `http://www.mdsplus.org/old/javascope/ReadMe.html`. Last accessed 29 January 2004.
13. Reid, J.: The future of fortran. Computing in Science and Engineering **5** (2003) 59–67

Task Modeling in Computer Supported Collaborative Learning Environments to Adapt to Mobile Computing

Ana I. Molina, Miguel A. Redondo, and Manuel Ortega

Dpto. de Informática. Universidad de Castilla – La Mancha
Paseo de la Universidad, 4. 13071 – Ciudad Real. Spain.
aimolina@inf-cr.uclm.es
{Miguel.Redondo,Manuel.Ortega}@uclm.es

Abstract. Using the new wireless technologies, mobile devices with small displays (handhelds, PDAs, mobile phones) are present in many environments. We are interested in the effective use of such ubiquitous computing devices for collaborative learning. We show here their application to a case study, the teaching of Domotics. To achieve our goal, we analyze the tasks which are susceptible of improvement through ubiquitous computing. We intend to identify common high-level task patterns in *Computer Supported Collaborative Learning* (CSCL) environments and guidelines that facilitate the creation of a complete semi-automatic environment that generates CSCL and ubiquitous tools, independent of the study domain and of the platform.

Keywords: Mobile computing, CSCL, PDA, automated generation of user interfaces, task modeling.

1 Introduction

The main goal of this article is to incorporate the ubiquitous computing paradigm in the teaching and learning of domains with a high experimental degree in order to take into account mobile computing possibilities [1, 2, 3]. Also, the features of these domains provide an excellent framework to analyze the collaborative process. Thus, we are going to study the methods that allow us to systematize these tasks. We will take as a starting point a collaborative e-learning environment based on the desktop metaphor, following the so-called "Domosim-TPC" [1]. To achieve our goal, we analyse the tasks (already modelled in the aforementioned system) which are susceptible of improvement through ubiquitous computing. Once these tasks have been defined, we will develop a flexible architecture that will support them and will be extensible and applicable to other situations and necessities [4, 5]. With this architecture we will implement a prototype materialising the theories outlined. The prototype will be applied to the learning of Domotics and integrated in the Domosim-TPC[1] environment.

In this paper the modeling of the main tasks in Domosim-TPC are shown. This is necessary to adapt the interface to mobile computing support. Also, it is used for automating this process. First, ubiquitous computing concept is introduced; next, we

[1] En Web, http://chico.inf-cr.uclm.es/domosim

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 786–794, 2004.
© Springer-Verlag Berlin Heidelberg 2004

describe the main features in the Domosim-TPC tool (used as a starting point environment) and some ideas about automated generation of user interfaces and task modelling. In the following section, the stages necessary to develop a ubiquitous version of the aforementioned system are enumerated. And finally we will describe the evolution process of the asynchronous tools in Domosim towards PDA support (individual workspace to design models) and we will draw some conclusions.

2 Incorporating the Ubiquitous Computing Paradigm in the Teaching and Learning of Domains with High Experimental Degree

2.1 Domosim-TPC

The domain where our investigation is being applied is the learning of the design of automated control facilities in buildings and housing, also called Domotics. The term Domotics is associated to the set of elements that, when installed, interconnected and automatically controlled at home, release the user from the routine of intervening in everyday actions and, at the same time, provide optimized control over comfort, energetic consumption, security and communications. In this kind of training, the realization of practical experiments is specially important. In order to soften this problem by means of the use of technology, we have developed a distributed environment with support for distance learning of domotics design: DomoSim-TPC.

In DomoSim-TPC the teacher carries out a presentation of theoretical contents. Next, the students are organized in small groups whom the teacher assigns the resolution of design problems. The students use an individual workspace to design the models that they consider will satisfy the requirements of the proposed problems. Later on, they discuss, comment and justify the design decisions taken, building a base of shared knowledge.

2.2 Ubiquitous Computing in Education

Ubiquitous computing as the interaction paradigm was first introduced by Mark Weiser [6, 7] in Xerox PARC laboratories in Palo Alto in 1991. This new paradigm changes the concept of using the computer by distributing multiple low-powered computers or small computer components throughout the environment, trying to hide their presence, concealing their use by disguising their appearance adapting them to the traditional tools used in the classroom. From the learning perspective, we follow the theory of collaborative learning. We try to fuse the principles of the CSCL and ubiquitous computing. We consider a classification of systems according to the characteristics of ubiquity involved and to the kind of collaboration adopted. Some of these systems belong to a category that supports collaboration in asynchronous discussion interfaces. This type of systems provides a set of tools for discussion making it possible to present individual work to the group. These systems define both an individual workspace and a group workspace for discussion, and even in some cases areas of results, but only of an asynchronous type.

The activities that the students carry out in Domosim-TPC, our CSCL design tool, consist of two stages:

- Specification of models and planning of their design strategy. In this stage the students, in an individual way, reflect and plan the steps to build a model satisfying the requirements proposed in the problem formulation. The strategy traced by the user is dynamically contrasted with an optimal plan of design for this problem.
- Discussion, argument and search of agreement in the characteristics of the models individually built. In this stage, the participants discuss about the models built, about their types and about the steps carried out to obtain them. From this process a proposal (model) is obtained, reflecting the view point of each participant.

2.3 Towards a Ubiquitous Domosim

Our purpose entails improving the traditional classroom environment with the collaborative and the ubiquitous computing paradigms. We intend to implement a prototype in a particular environment for collaborative learning (Domosim-TPC). We have to adapt the asynchronous tools of Domosim to the characteristics of mobile devices. To do this, it is necessary to restructure the user interface to adapt it to the constraints of size and utility of this kind of appliances.

In order to sustain a learning activity, we identify the concept of space. This is, a virtual structured place with resources and tools to perform a task (to solve a problem). There are three spaces:

- *An individual workspace to design models satisfying a specification.*
- *A shared space for discussion and argumentation* about the design actions that the learners have planned. This workspace provides support for issue-based conversation and relates the dialogue contributions with the actions of design models.
- Another *shared space with the results of the discussion process*. In this workspace the table of contents metaphor is used. This table contains and organizes design models in which the learners have reached agreement.

If we want to generalize this process to the learning environment of other disciplines, we can automate the transformation process of the user interface. For CSCL tools developers, this introduces the problem of constructing multiple versions of applications for different devices. There are many dimensions to consider when designing context-dependent applications (environments, platforms, domain, users,…). We intend to identify similar tasks in CSCL tools to automate the attainment reach of a ubiquitous version of a collaborative design environment.

2.4 Analyzing Task-Based Automated Generation of User Interfaces

There are several solutions to the problem of building device-independent user interfaces. An interface model for separating the user interface from the application logic and the presentation device is necessary. There are several markup languages that help in this purpose (UIML [8], XML,…). This kind of languages allows the

production of device-independent presentations for a range of devices. But these solutions do not provide high-level guidance guaranteeing quality across multiple versions of applications. We propose the use of a model-based design of GUI [9], which focuses on the tasks supported. The idea is that task analysis provides some structure for the description of tasks or activities, thus making it easier to describe how activities fit together, and to explore what the implications of this may be for the design of user interfaces. A number of approaches to task modeling have been developed (GOMS [10], HTA [11], CTT [12, 13],…). The logical decomposition of tasks is reflected in the selection, consistency and grouping of elements in the GUI obtained.

We intend to identify *common high-level task patterns* in CSCL environments that allow the development of a complete semi-automatic environment that generates CSCL and ubiquitous tools, independent of the study domain and the platform.

We use graphical *ConcurTaskTrees* (CTT) notation [12, 13] for analysing tasks in Domosim-TPC. Some important features are supported in CTT: hierarchical logical structures, temporary relationships among tasks, and cooperative tasks modelling. Cooperative work combines communication, individual action, and collaboration. This notation aims to provide an abstract representation of these aspects.

The new context of use implies reconfigurations of the UI that are beyond he traditional UI changes [14, 15], such as the redistribution of widgets across windows or tabs in a tabpanel, the reduction of a full widget to its scrollable version, without using a sophisticated widget, to replace a interactor with a smaller alternative, etc. The technique of automatically selecting an appropriate interactor while considering screen resolution constraints has already been investigated and shown to be feasible [16].

The process of generating a user interface in a model-based system can be seen as that of finding a concrete specification given an abstract one (*the mapping problem*) [17]. Once the elements of the abstract task of the user interface have been identified, every interactor has to be mapped into interaction techniques supported by the particular device configuration considered (operating system, toolkit, etc.)

The success of model-based systems has been limited. Several systems have attempted to automatically generate user interfaces in a model-based environment (UIDE [10], Mecano [18], Trident [19],...). The idea of these systems was to try to automate as much as possible the interface generation process from a task model, but these are very closed in specific domains.

2.5 Stages in Evolution Process

The process of evolution of Domosim-TPC towards ubiquitous computing consists of several stages:

a) *Analysing tasks that can be improved by using ubiquitous computing.*
b) *Design of tasks* taking ubiquitous computing paradigm principles into account. Modeling and the design of certain tasks must be reconsidered. The devices and protocols necessary for materializing these tasks must be decided.
c) *Implementing a prototype* that applies proposed theories.
d) *Evaluating the prototype in real contexts.*

e) *Identifying the task patterns that could be common in CSCL environments,* based on the resolution of proposed problems and simulation of solutions contributed by students.
f) *Creating a tool that allows*, from a tasks model of a CSCL application, *obtaining in a semiautomatic way the equivalent interface for several mobile devices*.

At the moment, we are in stage c. We are developing the asynchronous tools in Domosim-TPC for PDA.

2.6 An Example of Generation of the User Interface for PDA

In this section, we describe the analysis of the main tasks in asynchronous workspace of Domosim-TPC. This analysis should be done at low level. It has to determine the kind of interaction task (for example, enter text, select a Boolean value, select a numeric value) and the kind of domain application objects manipulated by the tasks. This information facilitates the identification of the visual component (*widget*) that best allows the realization of a particular task, taking target device restrictions into account.

2.6.1 Obtaining the Interface for PDA of the Individual Planning Space

We intend to obtain the ubiquitous version, and in particular, for PDA, from the tasks analysis of the individual plan edition space in Domosim-TPC. Figure 1 shows the user interface of the plan editor. This is structured in separate areas: the problem formulation, the list of tasks to realize (tasks which give structure to the problem), the icon bars representing design actions/operators, the sequence of design actions already planned, the current action under construction and a set of buttons dedicated to supporting several general functions. The design actions that the student should choose are displayed in the user interface by means of icons in toolbars. They are grouped in four categories according to the components of an action: the kind of action, the management area, the housing plan and the domotics operator.

In figure 2 we can see the task model in CTT notation for individual planning. Figure 3 gives details about the abstract task *PLANNING*.

To obtain the version for PDA of the individual workspace, temporary relationships among the tasks and the domain application objects manipulated to perform them must be taken into account. This information allows creating the interface in which both the widgets (user interface objects) that show domain application objects (internal objects) and the widgets that allow executing certain actions applicable to these internal objects must appear together. In the editor of plans of design (the individual workspace) two internal objects are handled: the *design action* and the *design plan* (a collection of design actions). In figures 2 and 3 the names of both objects are written in uppercase. They are part of the name of the tasks that manipulate them.

Diagram in figure 2 shows the general functions that can be performed on the design plan. It can be shown graphically. There are two modes of *visualization*: a list of nodes (a node represent a action) connected by arrows (that represent precedence relationships); and the design of the scene that is created for executing the planned

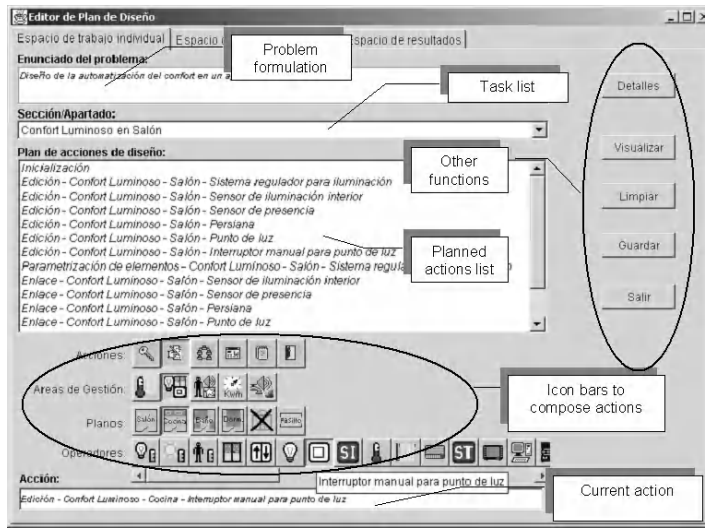


Fig. 1. Plan editor user interface

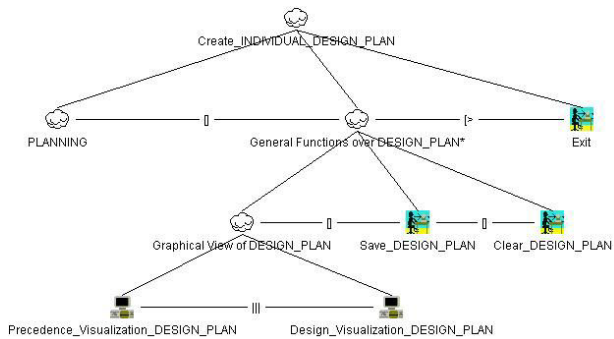


Fig. 2. Tasks modeling of space for individual planning in Domosim-TPC

actions list. Also we can *save* the design plan. The option *Clear* eliminates all the information contained in the actions list. These actions are applicable to the object *plan design*. These must appear in the user interface next to the object related (the list box that shows the sequence of steps in the plan). The resulting interface for PDA of these subset of tasks are shown in figure 4 (a).

In addition, the individual plan editor handle objects *design action*. In the diagram shown in figure 3 the actions *Add_DESIGN_ACTION* and *Delete_DESIGN_ACTION* are included. The first one has certain complexity. When a task (that means an operation over a internal object) is of the interaction type, the mapping to a perceptible object (a widget in the interface) is more direct. This kind of operations can be represented by means of buttons, options in a menu or a contextual menu. It has been applied to the mapping of the operation *Delete_DESIGN_ACTION*, or the aforementioned generic functions, which the user can perform on the object *DESIGN_PLAN*.

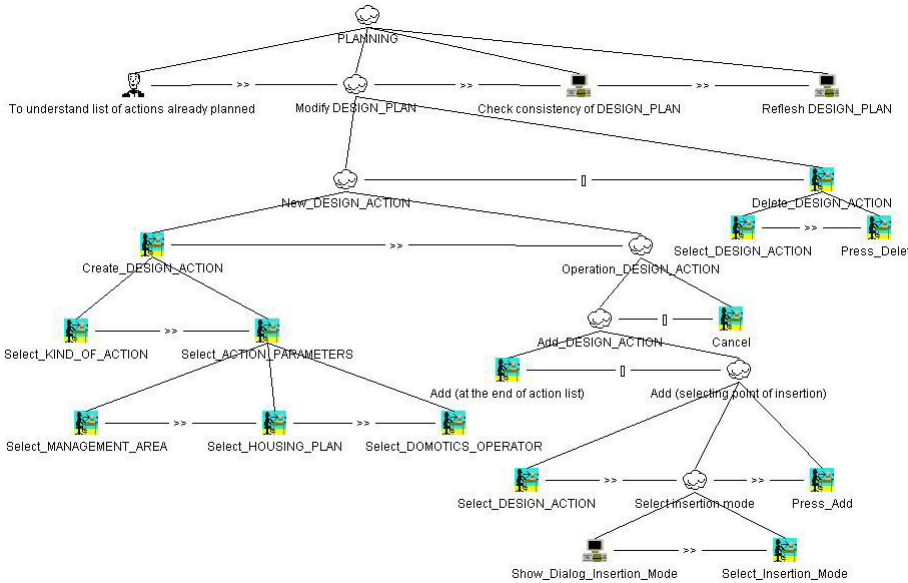
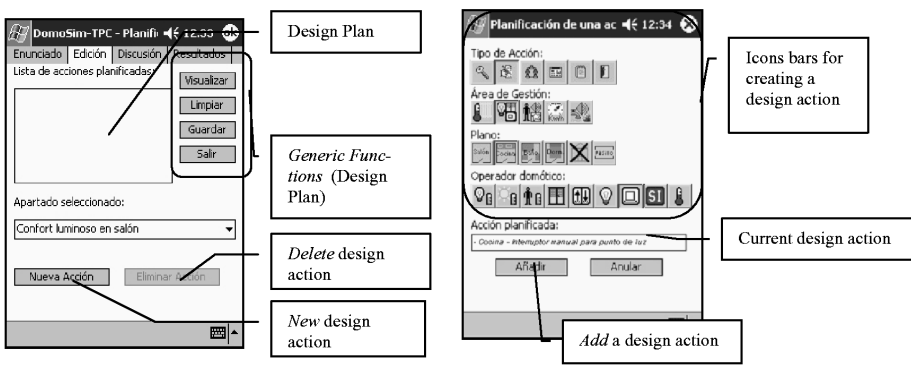


Fig. 3. Modeling of abstract task *PLANNING*



(a) (b)

Fig. 4. PDA Version of the interface of Domosim-TPC. (a) Interface that allows showing and performing actions on the *DESIGN_PLAN*. (b) Dialog box that allows the creation of a new *DESIGN_ACTION*.

However, when a task has a certain complexity, i.e., when a task is represented by an abstract task, with several abstraction levels and several interaction tasks (this occurs in the task *New_DESIGN_ACTION*), more complex visual components are necessary (a panel, in a PC version of the interface; or in a PDA, where there are display resolution constraints, a dialog box is a better choice). This occurs in the task that allows creating new design actions, as we can see in figure 4 (b). This dialog box appears whenever a new design action is created.

3 Conclusions and Future Works

Using the new wireless technologies, mobile devices with small displays (handhelds, PDAs, mobile phones) are present in many environments. We are interested in the effective use of such ubiquitous computing devices for collaborative learning. We show its application to a case study, the teaching of Domotics. To achieve our goal, we analyze the tasks which are susceptible of improvement through ubiquitous computing. We intend to identify common high-level task patterns in a CSCL environment and guidelines that permit the creation of a complete semi-automatic environment that generates CSCL and ubiquitous tools, independent of the study domain and of the platform.

We take as a starting point a collaborative e-learning environment of domotics design, based on the desktop metaphor, called Domosim-TPC. We intend to adapt this tool to the characteristics of mobile devices. To do this, it is necessary to restructure the user interface to adapt it to the constraints of size and utility of this kind of appliances. We use graphical ConcurTaskTrees (CTT) notation for analyzing tasks in the aforementioned system. Some important features are supported in CTT: hierarchical logical structures, temporary relationships among tasks, and cooperative task modeling. Cooperative work combines communication, individual actions, and collaboration. This notation aims to provide an abstract representation of these aspects. The logical decomposition of tasks is reflected in the selection, consistency and grouping of elements in the GUI obtained. We intend to automate the generation of the mobile equivalent interface for a desktop CSCL version.

Acknowledgments. This work has been partially supported by the Junta de Comunidades de Castilla – La Mancha and the Ministerio de Ciencia y Tecnología in the projects PBI-02-026 and TIC2002-01387.

References

1. Redondo, M.A., Planificación Colaborativa del diseño en entornos de simulación para el aprendizaje a distancia. Tesis Doctoral. Departamento de Informática. Universidad de Castilla-La Mancha. 2002
2. Soloway, E., Norris, C., Blumenfeld, R., Fishman, B., Krajcik, J., & Marx, R. Log on Education: Handheld devices are ready-at-hand. *Communications of the ACM*, 44 (6) 15–20
3. Soloway, E., Grant, W., Tinker, R., Roschelle, J., Mills, M., Resnick, M., Berg, R. & Eisenberg, M. *Science in the Palms of their Hands*. Communications of ACM, August 1999, 42-8, 21–26.
4. Ortega, M. , Redondo, M.A., Paredes, M., Sánchez-Villalón, P.P., Bravo, C., Bravo, J., Ubiquitous Computing and Collaboration: New Paradigms in the classroom of the 21st Century., Computers and Education: Towards a Interconnected Society, M. Ortega and J. Bravo (Eds.), Kluwer Academic Publishers , pp. 261–273, 2001.
5. Ortega, M., Paredes, M., Redondo, M.A., Sánchez-Villalón, P.P., Bravo, C., Bravo, J.; AULA; A Ubiquitous Language Teaching System, Upgrade Vol. II, Nº 5, pp. 17–22. 2001.
6. Weiser, M. The future of Ubiquitous Computing on Campus, Comm. ACM 41-1, January 1998, 41–42. 1998.

7. Weiser, M. (1991) *The computer for the twenty-first century*. Scientific American, September 1991, 94–104.
8. Abrams, M., Phanourious, C., Batongbacal, A.L., Williams, S. & Shuster, J. UIML: An appliance-independent XML user interface language. In A. Mendelson, editor, *Proceedings of 8th International World-Wide Web Conference WWW'8 (Toronto, May 11-14, 1999*), Amsterdam, 1999. Elsevier Science Publishers. Accesible at http://www8.org/w8-papers/5b-hypertext-media/uiml/uiml.html.
9. Paternò, F. Model-Based Design and Evaluation on Interactive Applications. Springer Vergal, ISBN 1-85233-155-0 (1999)
10. Foley, J., et al., UIDE-An Intelligent User Interface Design Environment, in Intelligent User Interfaces, J. Sullivan and S. Tyler, Editors. 1991, Addison-Wesley. p. 339–384.
11. Annett, J., Duncan, K.D., *Task Analysis and Training Design,* Occupational Psychology, 41, pp. 211–221, 1967.
12. Paternò, F., Mancini, C. & Meniconi. ConcurTaskTree: A diagrammatic notation for specifying task models. In S. Howard, J. Hammond, and G. Lindgaard, editors, *Proceedings of IFIP TC 13 International Conference on Human-Computer Interaction Interact'97 (Sydney, July 14-18, 1997)*, pages 362–369, Boston, 1997. Kluwer Academic Publishers.
13. Paternò, F., Santoro, C. & Tahmassebi, S. Formal model for cooperative tasks: Concepts and an application for en-route air traffic control. In P. Markopoulos and P. Johnson, editors, *Proc. of 5th Int. Workshop on Design, Specification, and Verification of Intractive Systems DSV-IS '98 (Abingdon, June 3-5 1998*), pages 71–86, Vienna, 1998. Springer-Verlag.
14. Eisenstein, J., Vanderdonckt, J. & Puerta, A. Adapting to mobile contexts with user-interface modeling. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications WCSMA'2000 (Monterey, December 7-8, 2000)*, pages 83–92, Los Alamitos, 2000. IEEE Pres.
15. Eisenstein, J., Vanderdonckt, J. & Puerta, A. Applying model-based techniques to the development of user interfaces for mobile computers. In *Proceedings of ACM Conference on Intelligent User Interfaces IUI'2001 (Alburqueque, January 11-13, 2001*), pages 69–76, New York, 2001. ACM Press.
16. Eisenstein, J. & Puerta, A. Adaptation in Automated User-Interface Design. *Proc. of IUI'2000* (New Orleans, 9-12 January 2000), ACM Press, New York, 2000, pp. 74–81.
17. Puerta, A. & Eisenstein, J. Towards a General a Computational Framework for Model-Based Interface Development Systems. *Proc of IUI99: International Conference of Intelligent User Interfaces*, Los Angeles, January 1999, in press.
18. Puerta, A. R. The MECANO Project: Comprehensive and Integrated Support for Model-Based Interface Development, in *Proc. of CADUI96: Computer-Aided Design of User Interfaces.* 1996. Numur, Belgium.
19. Vanderdonckt, J.M & Bodart, F. Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection, in *Proc. of InterCHI'93*. 1993: ACM Press.

Computational Science and Engineering (CSE) Education: Faculty and Student Perspectives*

Hasan Dağ[1], Gürkan Soykan[1], Şenol Pişkin[1], and Osman Yaşar[2]

[1] Computational Science and Engineering Program, Istanbul Technical University
Maslak-Istanbul, Turkey
dag@be.itu.edu.tr
[2] Department of Computational Science, State University of New York,
College at Brockport, Brockport, NY 14420
oyasar@brockport.edu

Abstract. Computational Science and Engineering is a multi-disciplinary area with growing recognition in the past decade. The affordability and availability of high performance computers and communication have helped shrink the technological divide; both internally within each country and globally among industrial and developing countries. Although there is no standard or accredited curriculum in this multi-disciplinary field yet, there are a number of departments, institutes, programs that have adapted variations of a sound education. Several programs have been established in Turkey, following the lead of "Computational Science and Engineering" (CSE) graduate program at Istanbul Technical University. In this article, we will discuss the establishment, development, current state and future plans of the CSE program at Istanbul Technical University. Faculty and student perspectives are presented at the same time so the reader is exposed to all potential issues.

Keywords: Computational Science and Engineering Education, Curriculum

1 Introduction

CSE education is now becoming widespread all over the world. It initially started as an overlap of computer science, mathematics, and science/engineering [1]. Over the years, however, it has evolved into a discipline of its own with a broad platform for both research and education [2]. It has two main characteristics: 'science and engineering of computation' and 'science and engineering, done computationally.' Educators are also using integrative nature of CSE to explore it as an inquiry-based, project-oriented, and constructivist pedagogy, whose impact goes beyond its disciplinary nature [3].

Istanbul Technical University (ITU) has long recognized the importance of CSE and designed a master and doctoral program under the Informatics Institute in 2000. ITU is the oldest technical university in Turkey with its foundation dating back to

* Authors acknowledge support by Turkish State Planning Organization (www.dpt.gov.tr), National Science Foundation, The State University of New York (www.suny.edu), and College at Brockport (www.brockport.edu).

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 795–806, 2004.
© Springer-Verlag Berlin Heidelberg 2004

1773. The university currently has around 6000 graduate students and 11000 undergraduate students. It is one of the three top technical state universities in Turkey. The Informatics Institute was established in 1999 and it currently has 4 graduate programs (3 master's and doctoral programs and one master's only program) and 4 professional master's program. The CSE program is supported by instructional capacity of Institute's core faculty, whereas the remaining graduate programs receive instructional services from other academic departments as well. In this way, the institute behaves both as an academic center and an administrative office for graduate studies in informatics.

Historically, most of the CSE programs have been formed in two ways, either as a 'track' residing within an existing department (mathematics, computer science, or physics) or a specialization within a partnership of multiple departments [4-7]. In the latter, a set of CSE courses complement required courses for the main degree. Students earn a degree in their home department with a specialization or advance certificate in CSE. In recent years, there have been several standalone programs, most notably the Computational Science Department at SUNY Brockport whose chair (also an author here) has visited ITU several times. A close collaboration exist today between ITU and SUNY, which has recently resulted in a new project from Turkish State Planning Agency to establish a national supercomputer center in Turkey.

In this article, the establishment, development, current state, and future plans of the Computational Science and Engineering program of Informatics Institute of Istanbul Technical University are explained. To provide a well-grounded feedback to others considering to start similar programs, we are presenting perspectives of our core and visiting faculty and views of the student body on curriculum, courses, infrastructure and the future plans of the CSE program.

2 CSE Program at ITU

Being the first such program in Turkey, the ITU CSE program is unique in the world with respect to its structure. Many graduate programs draw instructional and research capacity from a variety of departments on their campuses. That is, in addition to a few core CSE courses, the rest of required or elective courses are taken either from a science or engineering department. The program at ITU was designed to offer enough core and elective course offerings within the program itself so students could spend their time together within the center. In addition to necessary classrooms and computer labs, office rooms were allocated for most of the graduate students. Several highlights of the program are:

- In consistent with latest developments around the world and as a result of input by the international advisory board, the curriculum has been updated 3 times.
- The faculty's backgrounds are civil, electrical, geo-physics, naval and oceanography engineering, math, chemistry, and physics. With an international faculty exchange program, the faculty capacity has been expanding further.
- The program is open to all students with an undergraduate degree. Some students may be required to take 1-2 semesters of undergraduate courses, which do not count towards required credits.

- Master's students are required to take a minimum of 8 courses (3 credits each, total 24 credits) and up to 2 semesters of thesis work. For Ph.D. students, there is additional coursework (24 credits) before the qualifying examination. After passing their qualifying examinations, Ph.D. students go on with their thesis research for up to 4 semesters.
- Through support from the State Planning Organization (DPT) of Turkey, we are able to bring experts from all over the world for teaching, supervising graduate students, and designing workshops etc. Long-term visiting appointments are available for international faculty who provide formal supervisory support (as co-advisor) to graduate students. Support is also available for graduate students to go abroad for duration of 6-10 month for research to work with their international advisors.

2.1 Research Areas of the Core Faculty

There are currently 3 faculty members who devote all of their time to the program both in teaching and research. In addition, there are a few other faculty members who are involved in teaching and advising students. Although there have been skeptics about our program's sustainability over long term, the number of associated faculty from other departments is growing. After 3 years of experience, there seem to be a considerable optimism on the campus towards the CSE program. Several new groups, such as computational physics, chemistry, and biology, have formed recently as a result of program's outreach effort. The following is a list of research topics and projects undertaken in our program.

A. Computational Fluid Dynamics Group (Led by Serdar Çelebi):
- Blood Flow Modeling for Carotid Artery Bifurcation
- Liquid Sloshing
- Nonlinear Unsteady Ship Hydrodynamics
- Computational Free Surface Hydrodynamics
- Nonlinear Wave Body Interactions
- Numerical Wave Tank Simulation
- Floating Body Motions and Fluid-Structure Interactions

B. Computer Aided Geometric Modeling

C. Parallel Algorithms & Grid Computing

D. Applied Math & Computational Tools (Led by Hasan Dağ):
- Large Scale Sparse Matrix Techniques
- Parallel Iterative Methods
- Preconditioner Design for Iterative Methods
- Evolutionary Computing
- Stability Analysis
- Computational Finance

E. Molecular Optimal Control Theory Group (Led by Metin Demiralp):
- Optimal Control of Quantum Harmonic Oscillator
- General Reduction Methods for Field and Deviation Equations
- Stability and Robustness of Control Solutions

- Multiplicity and Uniqueness Issues in Quantum Optimal Control
- Applications of these subjects: Quantum Computation, Nano-technological

F. High Dimensional Model Representation Group

- New Approximation Schemes for Multivariate Functions
- Factorization Schemes
- Hybrid Schemes
- Weight Function Determination Based Schemes
- Random Data Partitioning
- Applications of these subjects: Multivariate interpolation, Quantum Mechanical and Statistical Mechanical Problem Solvers

3 Faculty Perspectives

The core faculty involved in the CSE program are from disciplines such as chemistry (molecular dynamics), electrical engineering, naval and oceanography engineering, and they all have strong background in computer science related topics such as programming languages, parallel programming, operating systems etc., and applied mathematics such as numerical methods, applied linear algebra etc. After several years of CSE experimentation, there is a realization in the program today that faculty's expectations from students may be too high. Most of the incoming students do not have enough math or computer science background.

The teaching load of the core faculty is too much. The support from ITU has not been enough. Bridge courses (Computational-X, where X is physics, chemistry, biology, etc) could enrich our program, but there are not enough incentives for disciplinary faculty to design and teach a CSE course, especially if this course is not a requirement in their own department. The attitude of the ITU faculty is that there is no need for a CSE program. They think most of their disciplinary courses already use computational tools.

Table 1. Number of accepted/graduated students

	Accepted		**Graduated**		**Left**	
Years	**MS**	**PhD**	**MS**	**PhD**	**MS**	**PhD**
2000-2001	24	10	-	-	15	5
2001-2002	29	7	-	-	10	2
2002-2003	13	10	12	-	3	4
2003-2004	14	14	5	-	-	1
Total	80	41	17	0	28	12

Since the nature of the program is of interdisciplinary, students come from many fields, including industrial, mechanical, electrical, civil and other engineering and science areas. It is almost impossible for such a diverse student body to have a common background in math and computer science. We have designed a certain number of non-credit courses to bring the students to a common starting point in math and computer science related topics before they embark their studies for a master or a PhD degree. The non-credit courses are mostly for master's students and it may take a year to complete them.

Table 2. Distribution of student areas in CSE

Area (undergraduate)	Number
Aeronautical Engineering	2
Astronautical Engineering	2
Chemistry	1
Civil Engineering	1
Computer Engineering	3
Geodesy and Photogrometry Eng	2
Food Engineering	2
Electrical Engineering	14
Industrial Engineering	4
Management	2
Mathematics	18
Mechanical Engineering	5
Medicine	1
Meteorological Engineering	1
Naval Engineering	2
Nuclear Energy	2
Ocean Engineering	1
Physics	9

Within the last three years, 80 masters and 41 PhD students have been accepted to the program and about a third of them have left the program without a degree, as shown in Table 1. Some of the reasons for high dropout are listed below.

- Lack of a serious commitment. A fraction of students attend graduate school in Turkey to avoid or delay military service. They register, but do not attend courses regularly for a couple of semesters until they are expelled.
- Rigor of the program. Due to strict regulations (high GPA) some students leave the program.
- Strong competitions leave some students behind. Especially those that do not come from highly ranked universities tend to give up. Students accepted to the program have a diverse background as shown in Table 2, and come from different universities, which obviously do not have the same teaching qualities. State universities cannot select their students. A centrally administered nation-wide entrance examination places the students according to students' choices and their exam results.

4 Student Perspectives

The survey conducted in the third year of the program show that students have the following perspectives:

Complaints:

- There are not enough application-oriented courses.
- There is a gap between courses, such that, while some of the courses are introductory-level whereas some other courses are quite advanced. As an example, the lack of an advanced programming and data structure course is a common complaint.
- The number of full-time faculty members is small.
- Co-advisorship is not encouraged strongly enough.
- Students are somehow anxious as to what their future job opportunities would be. This is due to the fact that CSE is not well known yet.

Praises:

- The support from the State Planning Organization of Turkey provides a good study and research environment. Due to this support, students have all types of computing and teaching facilities.
- Due to the support indicated earlier, program coordinators are able to invite very-good researchers/professors from abroad to come to teach and advise students. An example of that is the participation of Dr. Yasar, which prompted preparation of this article. He laid the first foundation of our program in 1998 by presenting the idea to our President.
- Program is updated almost yearly to enrich the content of the courses. Also, new courses are added to the program almost each year as the number of supporting faculty increases.
- Most of the students accepted to the program have Research and Teaching Assistantships besides a chance to stay in fully equipped dormitory rooms.

5 CSE Graduate Program

5.1 Computer Facilities

- **Student Lab-1:** 30 PCs with Intel Pentium 4-1500 Mhz, 128MBRam
- **Student Lab-**2: 20 PCs with Intel Pentium 2-Celeron 800Mhz
- **Student Lab-**3: 20 PCs with Intel Pentium Celeron 300 Mhz
- **Training Lab:** 20 Sun-Ray 150 Thin Client working alone hardware of a Sun-Blade 2000 explained below
- **Visualization Lab:**
  1. 5 Sun Blade Workstation with 900 Mhz Ultra Sparc III-Cu processor, 4 Gb memory, 21 Inch monitor
  2. 3 Sun Blade Workstation with 2-900 MHz Ultra Sparc III-Cu processor, 2 GBmemory, 21'' monitor
  3. 1 Power Mac G4 Workstation with two 1000-Mhz processor, 1 Gb memory, 19'' monitor
  4. A3 scanner & graph tablet
- Supercomputer Center:
  1. The main system is SunFire 12K high-end server. It gives researcher a shared memory environment. It has 16-900 Mhz UltraSPARC III Cu- CPU with 32 GB memory and 2TB disk storage, 4TB backup storage. With upgrades, another 16-1200 Mhz UltraSPARC III Cu-CPU with 32 GB memory was added.
  2. A Sun cluster with 24 units. (SunFire V210) Each unit has 2 -1200 Mhz UltraSPARC III Cu-CPU with 2 GB memory and 2x36GB disk storage.
- **Personal Computers:** 24 PCs with Intel Pentium 4 – 2.8 Ghz, 1 Gb RAM

5.2 Software

Software Packages:
- Matlab 6.5, Mathematica 5.0, Fluent 6.1, Gambit 2.1,
- Gaussian03, Star-CD 31.50, Sun One Studio 8.0,
- Kiva3, GMV, OpenDX, Paraview, GaussView

Free-for-educational-use software:
- ScaLAPACK, Octave, Mupad, GnuPlot

Operating systems:
- Solaris 8 and Solaris 9, Suse 8.0 and Suse 9.0, Mac-OS,
- Windows 2000server, Windows-XP

5.3 Master's Courses

HBM501B: *Fundamentals Software of Informatics:* Introduction to Operating System (Linux/Unix). Shell languages and programming with shell languages. The structure of makefile and make command. The structure of Network (ftp, ssh, telnet etc.). Word processing languages: Fundamental components of TEX, LaTEX, MuTEX and similar software. Fundamental commands of TEX language, its programming structure, tricky points of mathematical formulae typesetting. Subroutine facilities in TEX, examples.

HBM503B*: Introduction to Mathematical Methods in Computational Science and Engineering*: Vector Analysis: Properties of vectors, Gradient, divergence and rotation concepts, line and surface integral, integral theorem, orthogonal and curvilinear coordinate systems, introduction to ordinary diff. Equations, Laplace and Fourier Transforms. Series of Diff. Equations: Special functions, introduction to Boundary-Value Problems. Introduction to Complex Variables.

HBM511B: *Scientific Computation I*: Floating point representation, Taylor Series Expansion, Root Finding: Newton-Raphson, Secant, and Bisection Methods. Direct and iterative solution of linear and non-linear systems. LU and symmetric LU factorization. Complexity, stability and conditioning. Nonlinear systems. Iterative methods for linear systems (Gauss Seidel, Jacobi, SOR, CG etc.). QR factorization and least squares. Eigenproblems: Local and global methods. Introduction to numerical methods for ODEs.

HBM512B: *Scientific Computation II*: Polynomial forms, divided differences. Polynomial interpolation. Polynomial approximation: uniform approximation and Chebyshev polynomials, least squares approximation and orthogonal polynomials. Splines. B-splines and spline approximation. Introduction to numerical differentiation and integration. Introduction to numerical methods for solving initial and boundary value problems for ordinary differential equations.

HBM513B: *Parallel and Distributed Computing*: Fundamentals: Natural parallelism, evolution of parallel systems, multi processor systems, measuring program performance, and preparing for parallelism. Designing parallel algorithms: Methodical design, partitioning, communication, agglomeration, mapping, and case study. A quantitative basis for design: defining performance approaches to performance modeling, developing models, scalability analysis, evaluating implementations, I/O.

Table 3. Summary of CSE Program

Courses	Cr
Required	**12**
Fundamentals Software of Informatics	0
Introduction to Mathematical Methods in CSE	0
Scientific Computation I	3
Scientific Computation II	3
Parallel and Distributed Computing	3
Parallel Numerical Algorithms & Tools	3
Elective courses	**12**
Scientific Visualization	3
Numerical Discretization Techniques	3
Numerical Methods for ODE	3
Fund. of Optimization Theory & Applications	3
Computational Geometry	3
Computational Grid Generation	3
Optimal Control of Learning Systems	3
Numerical Solutions of PDE	3
Multi Variable Model Representation	3
Generalized Inverse Techniques in Engineering	3
Computational Nanostructure Physics	3
Perturbation Expansions & Their Applications	3
Fuzzy Decision Making Methods in CSE	3
Advanced Computational Methods for Fluids	3
Large Scale Sparse Matrix Computation in Engineering	3
Computational Complexity	3
Special Topics in Computational Sci. and Eng.	3
Total	**24**

HBM514B: *Parallel Numerical Algorithms & Tools*: Modular Design: design review, modularity and parallel computing, case study: matrix multiplication. Numerical libraries: The BLAS, implementation of BLAS, block algorithms, models for parallel libraries. Case study matrix factorizations, BLAS variants, parallel equation solution. Further Linear Algebra: QR factorization, iterative methods for linear equations, direct methods for sparse matrices, the linear least squares problem, eigenvalue/eigenvector problems. Other areas: Linear Multi Step methods. Tools: Open MP, C, F90, C++, High Performance Fortran, MPI, PVM.

HBM516B: *Scientific Visualization*: Data management: Data, metadata and common portable data formats. Basic concepts on perception ad color theory. Semiology of scientific communication. General design concepts for scientific illustration. A survey of visualization hardware. Taxonomy of graphics software tools and formats. Techniques for 2-D scalar and vector fields. Techniques for 3-D fields, volume rendering, ray tracing. Basics of animation. Virtual reality and VRML. Web integration of scientific information.

HBM517B: *Numerical Discretization Techniques*: Introduction to discretization techniques. Finite difference techniques. Finite volume techniques. Boundary integral techniques. All the above mentioned methods will be used to analyze 2 practical examples.

HBM519B: *Numerical Methods for ODE*: Initial-Value Problems (The Taylor Series Method, Finite Difference Grids and Finite Difference Approximations, Finite Difference Equations, Modified Euler Predictor-Corrector Method, Runge Kutta Method, Adams-Bashforth Method, Adams-Bashforth Moulton Method, The Modified Differential Equation, Multipoint Method, Nonlinear Finite Difference Equations, Systems of First-Order ODEs, Stiff ODE's Boundary-Value Problems, The Equilibrium Method, Other Boundary Conditions (Mixed Boundary Conditions, Boundary Condition at Infinity), Non-Linear Boundary-Value Problems (Newton's Method), Higher Order Methods.

HBM520B: *Fundamentals of Optimization Theory and Certain Applications:* Definition, classification, and the formulations of the optimization problems, convex sets and functions, conditions for local minima. Unconstrained optimization, steepest descent method, and conjugate gradient method. Newton Method and quasi Newton methods, convergence properties. Constrained optimization: review of linear programming and the simplex method. Nonlinear programming: Lagrange multipliers, Kuhn-tucker conditions, penalty and barrier functions, linear complementary problem and quadratic programming.

HBM597B*: Special Topics in Computational Science and Engineering*

5.4 PhD Courses

HBM601B: *Computational Geometry*: Introduction, parametric curve representation, arc length parameterization, the Serret Frenet Formulae, analytic representation of a curve, interpolation techniques, control polygon techniques: Bezier curves, B-Spline curves, rational B-Spline approximation (NURBS), curve generation, elementary mathematical properties of surfaces, parametric surface representation, the first and second fundamental form of a surface, the pde method for surface generation. Surface generation: Coons surface, Bezier surface, B-Spline surfaces, rational B-Spline (NURBS) surfaces, surface generation with volume constraints.

HBM602B: *Computational Grid Generation*: Preliminaries: The goals and tools of grid generation, mapping and invertibility, special coordinate systems. Structured curvilinear coordinates. Multiple methods of grid generation. Numerical implementation and algorithm development. Structured and unstructured grids. 3-D grid generation: Volume differential geometry, 3-D transfinite interpolation, 3-D Thompson-Themes-Mastin, 3-D Euler-Lagrange equations, Steger-Sorengon algorithm.

HBM603B: *Optimal Control of Learning Systems*: Recalls about optimization problems and Optimal Control Theory. Mathematical programming problems, mathematical economy related problems. The utilization of Lagrange multipliers. Kuhn-Tucker type constrained problems. Multi-stage deterministic allocation processes. Functional equations approach. The principle of the optimality and applications. Methods of the dynamic programming. Continuous deterministic processes. The properties of the cost functional. The bellman equation of the continuous systems. Time-optimal control problems. Synthesis problem. The classical methods. Generalized version of the Euler-Lagrangian equation for ordinary differential equation systems. Hamilton-Jacobi equation. Lagrange principle. Optimal control of linear building under seismic excitations. Instantaneous optimal control. Closed-loop and closed-open-loop controls. Kalman method. Iterative learning control methods. Linear optimal regulator problems.

HBM604B: *Numerical Solutions of PDE*: Elliptic Partial Differential Equations. Parabolic Partial Differential Equations, Stability Analysis. Hyperbolic Partial Differential Equations, Higher Order Schemes, Nonlinear Hyperbolic Schemes.

HBM605B: *Multi Variable Model Representation*: Multivariable functions and their various types of representations, Multivariable series expansions and their convergence properties, Clustering and Lumping techniques, Sobol expansion, High Dimensional Model Representation (HDMR), Estimations for truncation errors.

HDMR applications in partial differential equations: Applications in evolutionary systems, The contributions of the weight function in truncation error reduction, Applications in boundary value problems. Random variable utilization, Random variable methods in performing integration, Monte Carlo Method and its integration with HDMR, Applications.

HBM607B: *Generalized Inverse Techniques in Engineering*: Definition and importance of the inverse problem, inverse problems encountered in engineering. Construction of linear and nonlinear inverse models: inspection of the mathematical model, weighting the observations and the model parameters, importance of weighting, joint inverse models. Solution techniques: least-squares, most-squares, damped least-squares, eigen-value and eigen-vector analyses, Lanczos inverse. Interpretation of inaccurate, insufficient, and inconsistent data.

HBM608B: *Computational Nanostructure Physics*: A rapid review of some necessary topics in the courses Scientific Computation I and II, Adaptive Runge-Kutta Method and its physical applications, Numerical solution of Schrödinger equation in nanostructure physics. Data Analysis in nanostructure physics: curve fitting, spectral analysis. Numerical solutions of partial differential equations in nanostructure physics: Diffusion Equation, relaxation and spectral methods. Special functions and numerical integration in nanostructure physics. Various eigenfunctions, orthogonal polynomials, Bessel functions, Romberg iteration, Gauss quadrature, quantum perturbation and variation methods. Monte Carlo Method.

HBM609B: *Perturbation Expansions and Their Applications*: Perturbation concept, natural and artificial perturbations. Perturbation expansions, regular expansions, singular expansions. Perturbation in operators, Dependence of operators on perturbation parameter. Perturbation expansions of operators. Perturbation expansions in linear operators, convergence investigations, perturbation expansions in eigenvalue and inversion problems. Perturbation expansions with divergent but asymptotic behavior. Convergence acceleration in perturbation expansions. Divergence removal or convergence increase by using nonlinear approximants..

HBM610B: *Fuzzy Decision Making Methods in Computational Sciences*: Fuzzy Sets and Logic, Basic Definitions, Operations on Fuzzy Sets, Fuzzy Arithmetic, Basics of Fuzzy Decision Making, Aggregation Operators, Fuzzy Linear Programming, Fuzzy Approaches to Multiple Objective Programming, Applications to the Engineering Problems.

HBM612B*: Advanced Computational Methods for Fluids*: Introduction and governing equations. Generalized Curvilinear Coordinates (Geometry and coordinates, general review of vector and tensor analysis). Equations in Curvilinear Coordinates. Numerical Grid Generation. Reynolds-Averaged Transport Equations. Large Eddy Simulations (_filter functions, filtered equations, sub-grid models, dynamic SGS models). Incompressible Viscous Flows.

HBM614B: *Large Scale Sparse Matrix Computation and its Engineering Application*: Topics in engineering that require the solution of large sparse linear equations. Sparse matrices: storage schemes and operations, basic requirements for programming, Gaussian elimination for both sparse and dense matrices, permutation, factorization, and substitution phases of large sparse matrix computation. Sparse vector methods, Matrix re-factorization and matrix update. Special form of sparse matrices. Sparsity in nonlinear computation. Iterative methods. Comparison of direct and iterative methods. Eigenvalue and eigenvectors, generalized eigenvalue analysis.

HBM616B: *Computational Complexity*: Analysis of algorithms, Computational Complexity concept and fundamental definitions, distributions, probabilistic algorithms. General definition of recurrence relations, linear and nonlinear structures in first order recurrences. High order recurrence relations. Methods for the analytical and numerical solutions of recurrence relations. Binary divide - and - conquer type methods. Generating functions, ordinary and exponential generating functions. Expansions of generating functions, generating function based transformations. Counting methods by virtue of generating functions. Probability generating functions, bi-variate generating functions.

HBM697B: *Special Topics in Computational Science and Engineering*

6 Changes in the Future

- The program is scheduled to make new changes to deepen the knowledge level of students in engineering and science. There will be additional 6-9 credits in the application area, however, it will require to convince disciplinary ITU faculties to teach specially designed computationally courses.
- Since all research facilities are now in place, it is time to apply research and training skills to industry needs. Initial contacts on subjects such as “car crash testing”, “combustion simulations” have been made. The goal is to have at least 1-2 students working on industrial projects towards their MS degree.
- Difficulties continue for students to explain what CSE means to employers. Since the CSE program is a stand-alone program, a student diploma or transcript need to indicate the area of application. To do this we have to make a suggestion to the University Senate and have it approved.
- The number of faculty members must be increased. Toward this end, an assistant professor position is advertised and applications are being reviewed. Since the CSE is a new program and salaries in Turkey are not attractive in state-owned universities, it is hard to find qualified faculty members.

7 Conclusion

The ITU-CSE program was introduced with its problems and promises, including future developments plans. Both the faculty and student perspectives are given to make a comparison to the other CSE programs around the world. Even though there is still much work to do. Based on the success of our CSE program, the State Planning Organization of Turkey has already approved a follow-up project, with about $18-million budget, to set up a national High Performance Computing Center. All core CSE faculty members and Dr. Yaşar are Principal Investigators in this project, lending their expertise to carry the success of our CSE program into higher levels.

References

1. G. Golub and J. M. Ortega, Scientific Computing: an Introduction with Parallel Computing, Academic Press, Boston, 1993.
2. O. Yaşar, Computational Math, Science, and Technology: A New Pedagogical Approach to Math and Science Education, International Conference on Computational Science and Its Applications, 2004, Italy.
3. O. Yaşar and R. H. Landau, Elements of Computational Science and Engineering Education, SIAM Review, Vol. 45, No. 4, pp. 787–805, 2003.
4. SIAM Working Group on CSE Education, Graduate Education in Computational and Engineering, SIAM Review, Vol. 43, No. 1, pp. 163–177, 2001.
5. J. Seguel and D. Rodriguez, The Doctoral Program in Computing and Information Sciences and Engineering of the University of Puerto Rico, Future Generation Computer Systems, Vol. 19, pp. 1293–1298, 2003.
6. H.-J. Bungartz, A New CSE Master's Program at the Technische Universität München, Future Generation Computer Systems, Vol. 19, pp. 1267–1274, 2003.
7. O. Yaşar, et al., A New Perspective on Computational Science Education, Computing in Science & Engineering, Vol. 2, pp. 74–79 (2000).

Computational Math, Science, and Technology: A New Pedagogical Approach to Math and Science Education

Osman Yaşar*

Department of Computational Science
State University of New York, College at Brockport
Brockport, NY 14420
oyasar@brockport.edu

Abstract. We describe math modeling and computer simulations as a new pedagogical approach to math and science education. Computational approach to Math, Science, and Technology (CMST) involves inquiry-based, project-based, and team-based instruction. It takes the constructivist approach recommended by national learning standards. Our college has formed a partnership with local school districts to study impact of CMST on student achievement in math and science. We have trained more than 60 middle and high school teachers and teacher candidates. Preliminary results indicate that CMST-based professional development contributed an increase in passing rate (from 39% to 85%) of Rochester City School District in New York State high school math exam. This paper establishes relevant literature supporting CMST as an important scientific and educational methodology.

Keywords: Computational Math, Science, and Technology, Pedagogy, K-12 education

1 Introduction

The number of computer chips embedded in consumer products has quadrupled, reaching billions. There is a tremendous change in the workplace. Jobs have become technology-dependent and team-oriented. Employers are seeking for a flexible and multi-skilled workforce. However, at the same time, the number of people seeking an education in high-tech fields has dropped significantly in the past decades. Problem is tracked down to loss of interest and motivation in as early as secondary school years. Dramatic measures need to be taken to educate a future workforce capable of turning technological advancements into society's benefits.

Technology has dominated the workforce and our lives, for better or worse. However, it also offers remedies to deal with problems and shortfalls it has created. The literature contains extensive evidence that education can be considerably improved by focusing on higher-order cognitive skills using project- and inquiry-

* Author acknowledges support by National Science Foundation grant 0226962 (Math and Science Partnership).

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 807–816, 2004.
© Springer-Verlag Berlin Heidelberg 2004

based authentic learning, which is generally more effective than traditional didactic presentation in improving students' problem-solving skills. There is further evidence that technology applications can support higher-order thinking by engaging students in authentic, complex tasks within collaborative learning contexts. Computer technology offers tools to integrate mathematics and scientific inquiry within the same context. This integrated approach, namely CMST, employs math models to describe physical phenomenon, therefore bringing a new perspective to students about the usefulness of math as a tool in real life. It also complements traditional methods of performing science. Computer models enable one to perform sensitivity experiments, which are similar to laboratory experiments in the sense that one can perform controlled experiments. Computer simulations allow one to gain insights about real-life problems that are too complex to study analytically, too expansive to observe, and too dangerous to experiment. CMST enables teachers and students to create and navigate through visual representations of natural phenomena, therefore supporting inquiry-based learning while deepening content knowledge in math and science. Visualization allows rapid understanding of relationships and findings that are not readily evident from raw data.

This paper is an effort to convey current evidence on CMST approach. There is a significant research base supporting it as a new pedagogy. CMST not only offers a combined education to prepare the future workforce, but it also offers a layered-approach to gradually draw students into inquiring and learning more about math and science. It allows one to teach science using a deductive approach (from the general to the specific). At the outset, students are made aware that nature and its processes are governed by a handful of scientific laws. The degree of details and the number of mathematical steps necessary to represent most natural phenomena often cause students to perceive science as a complex discipline. CMST tools can be used to teach about a scientific topic via a series of student-controlled visual representations and simulations without having the students to know the mathematical and scientific details of the phenomenon under study. This provides a general simplistic framework from which one can introduce a topic and then move deeper as students gain higher interest that will help them move to more sophisticated levels of understanding. This motivational and layered aspect of technology use is a principal reason that educators strive to master and apply technology tools.

2 Evidence and Research Base

In 1989, the U.S. Office of Science and Technology Policy [1] challenged the educational system to (a) increase the supply of students seeking careers in science, technology, engineering, and mathematics disciplines, and (b) improve the scientific, mathematical, technological, and computational literacy of all students. Subsequently, the National Research Council, National Council of Teachers of Mathematics, National Science Teachers Association, and the International Society for Technology in Education each published a set of national K-12 educational content and professional development standards [2-4]. Many states, such as New York, also developed new Math, Science, and Technology (MST) standards for public schools. Common threads among these standards include the call for integration of technology into the curriculum, for classrooms to become learner-centered and inquiry-based and

Table 1. National Student Learning Outcomes relevant to integration of technology

National Science Education Standards	**National Council of Teachers of Mathematics Standards**	**Technology Foundation Standards**
Inquiry-6A: Ability to do scientific inquiry; understanding about scientific inquiry *Technology*-6E: Abilities of technological design, understanding about science and technology *Personal & Social Perspectives*-6F: Understanding of health, population growth, natural resources, environmental quality, global challenges *Program Standards*-7D: o Access to the world beyond the classroom o Formulate and revise scientific explanations and models using logic and evidence. Student inquires should culminate in formulating an explanation or model.	*Number and Operations*: o Understand numbers, ways of representing numbers, relationships among numbers o Compute fluently and make reasonable estimates *Algebra*: o Understand patterns, relations, and functions o Use mathematical models to represent and understand quantitative relationships o Analyze change in various contexts *Geometry*: Use visualization to solve problems *Data Analysis:* Develop and evaluate inferences and predictions based on data *Problem Solving*: Apply and adapt a variety of appropriate strategies to solve problems *Reasoning and Proof*: Make and investigate mathematical conjectures	o Use representations to model & interpret physical, social, and mathematical phenomena o Use technology for solving problems and making informed decisions o Use technology in development of strategies for solving problems in the real world

for all students to become critical-thinkers and problem-solvers. Tables 1–2 list national standards calling for integration of technology into math and science education.

As mentioned earlier, an *integrated* approach to MST has become necessary. Math modeling and computer simulations offer a way to apply math and computer skills to real-world applications. The computational approach to MST, or CMST, offers an understanding of science through the use and analysis of mathematical models on computers. Wright and Chorin [5] in their NSF report urged the creation of mathematical modeling courses for high school students. Thus, infusing modeling and technology into curriculum is consistent with national standards.

The traditional method of teaching science has a strong reliance on theory. On the other hand, course materials developed from CMST make science and math concepts more easily comprehensible to students, therefore significantly enriching the science curriculum. Science can be taught as a method of inquiry that incorporates facts as needed [6-10]. A high school program [11] reports that CMST approach led to a deeper understanding of science and math concepts and that doing science via computer modeling was more exciting than traditional classroom. Students appreciated teamwork. Teachers found the program revitalizing, bringing them a chance to explore new areas in technology, research, and pedagogy. Students and teachers claimed that they learned to approach problems in new ways and to develop new relationships among themselves in the process. They found new life in studying the science concepts and approaching their project with greater emphasis on the application of real life skills and research. They enjoyed the interactive, collaborative environment and the independence and responsibility the project generated.

CMST pedagogy takes the learner-centered or constructivist approach recommended by the standards. Constructivism holds that students learn better when

Table 2. National Teacher Preparation Standards calling for integration of technology into math and science education.

NCATE Standards in Science	**NCTM Standards in Mathematics**	**NCATE/ISTE Standards in Technology Education**
Content: Concepts & principles understood through science; concepts & relationships unifying science domains; processes of investigation in a science discipline; and applications of math in science research. **Nature:** Characteristics distinguishing science from other ways of knowing; characteristics distinguishing basic science, applied science, and technology; processes & conventions of science as a professional activity; and standards defining acceptable evidence and scientific explanation. **Inquiry**: Questioning and formulating solvable problems; reflecting on, and constructing, knowledge from data; collaborating and exchanging information while seeking solutions; and developing concepts and relationships from empirical experience. Computation & simulation complement theory & experiment as the 3rd way of scientific inquiry. **Skills of Teaching**: Science teaching actions, strategies & methodologies; interactions with students that promote learning & achievement; effective organization of classroom experiences; use of advanced technology to extend &enhance learning; &	**Grades K-12:** -Model, explain and develop computational algorithms -Use geometric concepts and relationships to describe and model mathematical ideas and real-world constructs -Collect, organize, represent, analyze, and interpret data -Identify, teach and model problem solving -Use a variety of physical and visual materials for exploration and development of mathematical concepts **Grades 5-12:** -Apply numerical computation and estimation techniques and extend them to algebraic expressions -Use both descriptive and inferential statistics to analyze data, make predications, and make decisions -Interpret probability in real-world situations, construct sample spaces, model and compare experimental probabilities with mathematical expectations, use probability to make predications -Use algebra to describe patterns, relations, and functions, and to model and solve problems -Understand calculus as modeling dynamic change, including an intuitive understanding of differentiation and integration and apply calculus concepts to real-world settings -Use mathematical modeling to solve real-world problems. **Grades 7-12:** -Understand the concepts of random variable, distribution functions & theoretical versus simulated probability and apply them to real-world situations -Have a firm conceptual grasp of limit, continuity, differentiation & integration, and a thorough background in the techniques & application of calculus -Have a knowledge of the concepts and applications of recurrence relations, linear programming, difference equations, matrices, and combinatorics	1. Use computer systems to run software; to access, generate & manipulate data; and to publish results; evaluate performance of HW/SW components of computers & apply basic troubleshooting strategies 2. Apply tools for enhancing professional growth & productivity; use technology in communicating, collaborating, conducting research, and solving problems; plan & participate in activities that encourage lifelong learning; promote equitable, ethical, & legal use of computer resources. 3. Apply computers and related technologies to support instruction in their grade level & subject areas; plan & deliver instructional units that integrate a variety of software, applications, and learning tools. 4. Professional studies in educational computing and technology provide concepts and skills that prepare teachers to teach computer/technology applications and use technology to support other content areas. 5. Apply concepts and skills in making decisions concerning social, ethical, and human issues related to computing and technology. 6. Integrate advanced features of technology-based productivity tools to support instruction. 7. Use telecommunications and information access resources to support instruction. 8. Use computers and other technologies in research, problem solving, and product development; use a variety of media, presentation, and authoring packages; plan and participate in team and collaborative projects that require critical analysis and evaluation; and present products developed. 9. Professional preparation in educational computing and technology literacy prepares candidates to integrate teaching methodologies with knowledge about use of technology to support teaching and learning. 10. Plan, deliver, and assess

&enhance learning; & the use of prior conceptions and student interests to promote new learning.	combinatorics -Use mathematical modeling to solve problems from fields such as natural sciences, social sciences, business, and engineering -Understand and apply the concepts of linear algebra -Identify, teach, & model problem solving	concepts and skills relevant to educational computing and technology literacy across the curriculum. 11. Demonstrate knowledge of selection, installation, management, and maintenance of the infrastructure in a classroom setting.

they are actively engaged in "doing," rather than passively engaged in "receiving" knowledge. Project-based learning is one way to create rich learning environments that invite students to construct personal knowledge and "authentic" learning. In addition to integrating the concepts of mathematics and science utilizing technology, the CMST pedagogy is based on the three key characteristics for constructivist learning environments as defined by Dunlap and Grabinger [12]: 1) generative learning, which requires students to become investigators, seekers and problem solvers; 2) anchored instruction which requires students to define the problem, identify resources, set priorities and explore alternative solutions, and 3) cooperative learning, which requires that students work in groups to tackle complex problems.

The types of learning that CMST can support are also listed in *Connecting the Bits* [13]. The CMST approach is both project- and team-based aimed at higher-order thinking skills. It is also learner-based and it supports authentic learning. As traditional, lecture-based classroom roles are changing, educators and students work collaboratively in more open-ended teaching and learning experiences. This combination of elements can transform uninvolved, at-risk students into active and invested learners.

While the constructivist approach to learning provides the framework, math modeling is the key element in the CMST pedagogy. In essence, the CMST approach is to gain an understanding of science applications through the use and analysis of math models on computers. Technology applications can support higher-order thinking by engaging students in authentic, complex tasks within collaborative learning contexts [14-15]. The action of integrating technology into the curriculum itself can be the impetus to creating a constructivist-learning environment. As Archer explains [16], "a constructivist approach toward learning, in which students work in rich environments of information and experience, often in groups, and build their own understandings about them – taps into the computer's greatest strengths." Wenglinsky [17] showed that computers used for real-world applications such as simulations or changing variables led to gains in student achievement. He analyzed the data from the mathematics portion of the 1996 National Assessment of Educational Progress given to 6,227 fourth graders and 7,146 eight graders. He found that a combination of project-based learning and technology resulted in achievement gains and that the effectiveness of computers in the classroom depended on how they were used. For 8th graders whose teachers had received sufficient professional development on computers, the use of computers to teach higher-order thinking skills was associated with a one-third of a grade-level increase in students' mathematics achievement [18]. It was also found that computers were more effective when used as a supplement to traditional instruction. Wenglinsky concluded that computers utilized in drill and practice had a negative effect on student achievement while computers used for real-

world applications such as simulations or changing variables were related to gains in student achievement.

A search of the literature identifies thousands of articles on classroom projects and the effectiveness of use of computers on learning. A meta-analysis [19] study found that computers are more effective when used in simulation or tutorial modes and enhanced student learning. A recent book, *Edutopia,* provides success stories for learning in the digital age [20]. Most of these reports can be considered testimonials, in which teachers tell how they use computer-based projects in their teaching. Benefits include:

- Increased motivation: Projects often report that students willingly devote extra time or effort to the project or that previously hard-to-reach students begin to participate in class.
- Increased problem-solving ability: Research on how to improve higher cognitive skills emphasizes the need for students to engage in problem solving and for teachers to provide specific instruction on how to attack and solve problems [21].
- Improved research skills: Computer technologies offer access to excellent sources of information. Students become independent researchers.
- Increased collaboration: Through collaborative projects, students gain experience in teaching their peers, evaluating the work of others, sharing information, and learning cooperatively. Current cognitive theories suggest that learning is a social phenomenon and that students will learn more in a collaborative environment.

Harvard researchers have studied ways to improve content, pedagogy, and assessment in education. Gardner's theory [22] emphasizes the need for personalization of schooling and education so a person can develop his or her own variation of multiple intelligences. Perkin's book [23] contains extensive evidence that education can be considerably improved by focusing on higher-order cognitive skills using project- and inquiry-based authentic learning, which is generally more effective than traditional didactic presentation in improving students' problem-solving skills.

Cooperative learning and collaborative problem solving frequently engage students as they work to complete a project. Cooperative learning has been shown to be effective, however it requires teachers to give students explicit training in collaboration and communication [24-25]. Project-based learning provides an authentic environment in which students can become more skillful at learning and problem solving through collaboration. Key haracteristics of project-based learning include:

- Students have some choice of topic and some control over the content of the project and the extent of their investigations. Students can shape their project to fit their own interests and abilities.
- The teacher acts as a facilitator, designing activities and providing resources and advice to students as they pursue their investigations. It is the students, however, who collect and analyze the information, make discoveries, and report their results.
- The context of the subject matter becomes larger than the immediate lesson.
- Students conduct research using multiple sources of information and the projects cut across multiple disciplines and a broad range of skills.

- The projects are based on teamwork. Individuals or small groups work on different components of a large task. Project members help each other close gaps and catch up to the progress of the overall teamwork.

An NFIE [13] article states that more than a decade of research, development, and implementation make it clear that integrating technology into the curriculum properly can produce dramatic change and improved prospects for at-risk students. Change comes about in part because effective use of technology for teaching dissolves many barriers and alters traditional methods and attitudes. New strategies are created. Successful technology integration involves complex sets of factors including, at a minimum, commitment to changing curriculum, high-quality professional development, flexible scheduling and instructional management, and a shift from rote learning to project-based learning. Change puts students at the center of their learning.

Another article published by the North Central Regional Educational Laboratory (NCREL) calls for "high standards and challenging learning activities for at-risk students." The CMST pedagogy is consistent with the NCREL approach. According to Means and Knapp [14], schools that fail to challenge at-risk students or encourage them to use critical thinking skills deprive them of creating a meaningful context for their learning. The CMST pedagogy gives all students opportunities to learn and employ mathematical and scientific concepts in the context of working on authentic tasks.

3 Recent Experience in CMST

The CMST approach has been adapted by many programs worldwide at the graduate and undergraduate levels [28]. Some of these programs have reached out to secondary schools [26-38]. After 4 years of implementing CMST at bachelor's and masters at SUNY Brockport, we started a Math and Science Partnership project in 2002 with Rochester City School District and Brighton Central School District to conduct the following CMST activities:

- A Joint Institute to coordinate meetings, activities, and development of new courses and challenging curricula using the CMST pedagogy.
- A summer institute to provide training to middle and high school teachers and college faculty. Teachers receive academic credits, stipend, and technology tools to enable them to extend CMST activities in their classrooms and school districts.
- A Scholarship opportunity for teachers and teacher candidates to pursue a BS or MS degree in computational science.
- A project-based Challenge program to promote collaborative work among project teachers, their students at grades 7-12, and college faculty mentors. Students to receive graphing calculators.
- A mentoring program to offer professional development to participating teachers through coaches at school districts and the college.
- Pedagogically improved courses at the college and in the school districts.
- Development and documentation of training materials, courses, and curricula.
- Dissemination of results and lesson plans to other teachers in the country.
- Testing new instructional technologies (hand-held devices and calculators).

- Development and administering evaluation instruments to measure student learning and teacher quality.
- Evaluation and analysis of targeted benchmarks by outside consultants.

In 2003, we trained more than 60 teachers and teacher candidates as well as 12 college faculty members. They received laptops, graphing calculators, and relevant software packages [39-41] to enhance teaching and learning of mathematics and sciences. Each Participant developed two lesson plans that involved use of CMST tools in classrooms. Hundred-percent of attendees indicated satisfaction with the training. While the input from the teachers indicated that they believed to have acquired intended knowledge, they were less confident about their level of skills in integrating it into their classrooms: 36% definitely felt prepared to apply modeling in their classroom, whereas 41% were "probably" prepared", 16% were unsure and 7% did not feel prepared. Interview data further revealed that teachers explained that their lack of confidence was due to their need for additional manipulation and experience with these new tools in the classroom. To support a continuous improvement, we trained more than 20 CMST coaches (among school teachers) who received advise from college faculty and provided similar training to other teachers in their districts after the summer institute. Monthly meetings with all involved teachers, faculty, and coaches provided demonstrations of CMST-based teaching in classrooms.

An independent consultant was employed to assess the impact of CMST-training on students learning. We received an invitation from U.S. House of Representatives to testify, on behalf of National Science Foundation, about our experiences on this project. Initial results indicate that student achievement (passing rate) in mathematics at Rochester City School District has gone from 39% to 85% since the beginning of our project. The Superintendent has credited part of this success to teacher training and summer programs. At the same time, the number of students seeking math and science fields at SUNY Brockport has also increased. The College moved from a Tier 3 to a Tier 2 category as a result of increase in percent (60%) of incoming freshman enrollment who have a high school average of 90 and/or SAT scores of at least 1200. This marks the culmination of a multiyear effort by Brockport to attract a higher caliber student body. The option of offering a Computational Science program in the college was an important factor in attracting high quality students. The CMST Institute offered scholarships to more than 20 students in 2003. The college also received NCATE accreditation [42] for its teacher programs. There were improvements to math and science education at the college through integration of CMST tools into 5 courses, including CPS 101 Introduction to Computational Science, NAS 401/501 Computational Approaches to Math, Science, and Technology Education I, NAS 601 Computational Approaches to Math, Science, and Technology Education II, ESC 350 Computational Methods in the Field Sciences**,** MTH 313 Mathematics for Elementary Teachers. More courses are expected to use CMST tools and approach in the 2003-2004 school year. The College Faculty Senate approved a combined BS/MS program in Computational Science. CMST-based training has been made part of initial ‘best practices’ recommendations in the College 5-year Strategic Plan.

4 Conclusion

Impact of educational technology has been demonstrated through extensive literature. Our own experience further demonstrates that CMST is an effective approach to conduct research and training in mathematics and sciences for faculty and students, both at the college-level and secondary schools. It has also demonstrated to raise motivation, interest, and curiosity among middle and high school students. Characteristics of the CMST approach include inquiry-based, project-based, and team-based instruction, and there is a significant research base supporting it as a new pedagogy. It offers the constructivist approach recommended by national and state learning standards. We will continue to implement and assess the impact of the CMST approach in the next several years. As more data becomes available, we hope that this study will offer a unique perspective to general public, academic institutions, and public schools about the role of computational science and technology education.

References

1. Office of Science and Technology Policy. (1989). The Federal High Performance Computing Program. President's Committee of Advisors on Science and Technology, Panel on Educational Technology. (1997). Report to the President on the Use of Technology to Strengthen K-12 Education in the United States. [On-line], Available: http://www.ostp.gov/PCAST/k-12ed.html.
2. International Society for Technology in Education. 2000, http://www.iste.org.
3. National Council of Teachers of Mathematics. (1989). Curriculum and Evaluation Standards for School Mathematics. (2000). Principles and Standards for School Mathematics. Reston, VA, http://www.ncate.org.
4. National Research Council. (1996). National Science Education Standards. Washington, DC: National Academy Press.
5. Wright, M. & Chorin, A. (1999). Mathematics and Science. National Science Foundation, Division of Mathematical Sciences.
6. Harrison, A. 1989. An exploration of the nature and quality of undergraduate education in science, mathematics, and engineering. Research Triangle Park, NC.
7. Johnston, K.L., and B.G. Aldridge. 1984. The crisis in science education: what is it? How can we respond? J. Coll. Sci. Teach. 14: 20–28.
8. Rutherford, F.J., and A. Ahlgren. 1990. Science For All Americans. Oxford University Press. NY.
9. Dunkhase, J.A., and J.E. Penick. 1990. Problem solving in the real world. J. Coll. Sci. Teach. 19: 367–370.
10. National Science Foundation. 1996. Shaping the Future: New Expectations for Undergrad Education in Science, Math, Engineering & Technology. NSF 96–139.
11. Porto, C. (1995). Pittsburgh Supercomputer High School Initiative http://www.supercomp.org/sc95/proceedings/568_CPOR/SC95.HTM
12. Dunlap, J. C. & Grabinger, S. (1996) Rich environments for active learning in the higher education classroom. In B. Wilson (Ed.), Constructivist learning environments: Case studies in instructional design (pp. 65–82).
13. National Foundation for Improving Education (2000). Connecting the Bits: http://www.nfie.org/publications/connecting.htm.
14. Means, B., & Knapp, M.S. (1991, January). Models for teaching advanced skills to educationally disadvantaged children. In B. Means & M. S. Knapp (Eds.), *Teaching advanced skills to educationally disadvantaged students.*

15. Means, B., Blando, J., Olson, K., Middleton, T., Morocco, C., Remz, A., & Zorfass, J. (1993). Using technology to support education reform, http://www.ed.gov/pubs/EdReformStudies/TechReforms/
16. Archer, J. (1998, October). The Link to Higher Scores. *In Technology Counts '98, a Special Report in Education Week on the Web.* [Online] Available at http://www.edweek.org/sreports/tc98/ets/ets-n.htm.
17. Wenglinsky, H. 1998. Does it compute? The relationship between educational technology and student achievement in mathematics. Princeton, N.J. ETS.
18. DeSessna, A. A. (2000) Changing Minds: Computers, Learning, and Literacy. MIT Press.
19. Bayraktar, S. (2002) *J. Research on Technology in Education,* 34 (2).
20. Chen, M. 2002. Edutopia: Success Stories for Learning in the Digital Age. Jossey-Bass.
21. Moursund, D. 1995. Increasing your expertise as a problem solver: Some roles of computers. Eugene, Oregon, ISTE.
22. Gardner, H. 1995. Reflections on multiple intelligences: Myths and messages. *Phi Delta Kappa* 200–209.
23. Perkins, D. 1992. Smart schools: Better thinking and learning for every child. New York: The Free Press.
24. Johnson, R.T. 1986. Comparison of computer-assisted cooperative, competitive, and individualistic learning. *Am. Educational Research Journal* 23 (3); 382–392.
25. Johnston, D. W. and R. T. Johnson. 1989. Social skills for successful group work. *Educational Leadership* 47 (4): 29–33.
26. Adventures in Supercomputing (AiS), www.krellinst.org
27. The ASPIRE program at Alabama, http://www.aspire.cs.uah.edu/.
28. Swanson, C., "Survey of Computational Science Education," www.krellinst.org.
29. The Maryland Virtual High School, http://destiny.mbhs.edu.
30. enVision for K-12 Curriculum, http://www.eot.org/projects/efk.html
31. Secondary Education in Computational Science, http://www.lcse.umn.edu/specs/
32. Nat'l Computational Sci Leadership Program, http://www.ecu.edu/si/te/profiles/
33. Homewood High School, http://199.88.16.12/compsci/compsci.html
34. National Computational Science Institute, http://computationalscience.net
35. Supercomputing Education Program, http://www.supercomp.org
36. REVITALIZE: http://www.eot.org/revitalise
37. The Shodor Foundation, http://www.shodor.org
38. The Krell Institute, http://www.krellinst.org
39. High Performance Systems, Inc. http://www.hpc-inc.com.
40. AgentSheets, http://www.agentsheets.com
41. MCS.Software, http://www.interactivephysics.com.
42. National Council for Accreditation of Teacher Education, http://www.ncate.org.

Resonant Tunneling Heterostructure Devices – Dependencies on Thickness and Number of Quantum Wells

Nenad Radulovic, Morten Willatzen, and Roderick V.N. Melnik

Mads Clausen Institute for Product Innovation, University of Southern Denmark, DK-6400 Sonderborg, Denmark
{radulle,willatzen,rmelnik}@mci.sdu.dk

Abstract. We present numerical results for GaAs/AlGaAs double-barrier resonant tunneling heterostructure devices. A particular emphasis is given to the influence of quantum well thickness and number of quantum well layers on current-voltage characteristic and carrier density profile. In the paper, we discuss results obtained for spatial dependencies of carrier densities, the peak and the valley current density, and corresponding potentials in N-shaped current-voltage characteristics for various resonant tunneling heterostructures. Results are based on the transient quantum drift-diffusion model. They are obtained by solving a coupled system of partial differential equations directly and, in contrast to previous analysis, no decoupling algorithms, procedures, or methods are used.

1 Introduction

Semiconductor devices that rely on quantum tunneling through potential barriers are playing an increasingly important role in advanced microelectronic applications, including multiple-state logic, memory devices, and high-frequency oscillators [1,2,3]. The local charge accumulation in quantum wells and nonlinear processes of charge transport across the barriers have been found to provide a number of mechanisms for Negative Differential Resistance/Conductance (NDR/C), bistability of the current at a given voltage, and nonlinear dynamics [4]. The N-shape of Current-Voltage Characteristics (IVC) may be adopted for realizing various logic functions.

By controlled layer-by-layer epitaxial growth of heterostructures in combination with lateral patterning, intricate artificial nanostructures with arbitrary shapes of barriers and wells can be designed and fabricated. Such bandstructure engineering can produce novel semiconductor devices with desired transport and optical properties. The aim of the current research activity in the field is not only to understand the complex and sometimes chaotic spatio-temporal dynamics of charge carriers in such structures, but also to make efficient use of those nonlinear transport properties in specific switching and oscillating electronic devices [4].

Physical device models are based on the physics of carrier transport, and can provide a great insight into the detailed operation of the device. In what follows, particular emphasis is placed upon low-dimensional GaAs/AlGaAs structures and the

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 817–826, 2004.
© Springer-Verlag Berlin Heidelberg 2004

nonlinear feedback between the space charges and the transport processes inherent in such structures. In Sect. 2, we present the physical model, as well as time and space discretization. Geometry and relevant parameters of the devices are mainly given in Sect. 3. A short description of numerical simulation is given in Sect. 4. Obtained results are presented and discussed in Sect. 5.

2 Theory and the Model

2.1 Origin and Validity of the Model

In the present paper, we employ the Transient Quantum Drift-Diffusion Model (TQDDM). It is a first-moment version of the isothermal Transient Quantum Hydrodynamic Model (TQHDM), where the velocity convection term is neglected [5]. The origin of the model is due to Ancona *et al*. [5,6]. The equation of state for the electron gas is generalized and density-gradient dependencies are included, which allow to account for quantum effects [5,6].

The TQDDM is limited to "high" temperatures ($T_0 \geq 77$ K) and "low" electron densities ($n \leq 3 \cdot 10^{19}$ electrons/cm^3), conditions often satisfied in semiconductor structure applications of most interest [7]. The density-gradient expansion is only valid if the coefficient ε, related to the quantum-correction term, is very small, $\varepsilon << 1$ [7]. Thus, the lowest-order density-gradient theory gives the best results when the characteristic length is large ($L > 10$ nm), and the effective mass close to its free-electron value [7].

2.2 The Transient Quantum Drift-Diffusion Model

A detailed derivation of the TQDDM is given by Pinnau *et al*. [8,9]. Basically, diffusion scaling is introduced in the TQHDM and the TQDDM is derived from a zero relaxation time limit.

The scaled TQDDM equations in 1D, stated on a bounded domain $\Omega \in R$, read [9]

$$n_t = \left(nF_x\right)_x \ , \tag{1a}$$

$$-\varepsilon^2 \frac{1}{\sqrt{n}}\left(\sqrt{n}\right)_{xx} + \log(n) + V = F \ , \tag{1b}$$

$$-\lambda^2 V_{xx} = n - C \ , \tag{1c}$$

where the dependent variables are: the electron density n, the quantum quasi-Fermi level F, and the electrostatic potential V. The time-independent doping profile C represents the distribution of charged background ions. The scaled Planck constant ε, the scaled Debye length λ, and the scaled relaxation time τ_0 (used in the numerical time discretization) are defined as

$$\varepsilon^2 = \frac{\hbar^2}{6m_e^* k_B T_0 L^2} \ , \ \lambda^2 = \frac{\varepsilon_s k_B T_0}{q^2 C_m L^2} \ , \ \tau_0^2 = \frac{k_B T_0 \tau^2}{m_e^* L^2} \ ,$$

where physical constants are: the reduced Planck constant $\hbar$, the Boltzmann constant k_B, and the elementary charge q. Physical parameters are: the effective electron mass m_e^*, the device operating temperature T_0, the permittivity ε_s, and the relaxation time τ, which depend on the material and the operating conditions of the device. The maximum absolute value of the doping profile C is denoted as C_m, and L is the characteristic (device) length.

The first term on the LHS of eq. (1b) is so-called quantum Bohm potential. The scaled current density, according to eq. (1a), is given by the following expression:

$$J = -nF_x \ . \tag{2}$$

Now, it is possible to introduce an external potential, modeling discontinuities in the conduction band, which occur in the resonant tunneling structures, and other semiconductor heterostructure devices [10]. For that reason, one must replace, in (1b), the potential V by $V + B$ [10], where B is a step function representing the nonnegative quantum well potential. The maximum value B_m, of the step-function B, depends on the content of Al in the ternary alloy composition. It is assumed that $B_m = 0.4$ eV, which corresponds to 65% Al in AlGaAs at 300 K [2].

In order to get a well-posed problem, the system of eqs. (1) has to be supplemented with appropriate boundary and initial conditions.

The electron density is assumed to fulfill local charge neutrality at the Ohmic contacts. Further, it is natural to assume that there is no normal component of the total current (including the quantum current) along the insulating part of the boundary. Finally, we require that no quantum effects occur at the contacts. These boundary conditions are physically motivated and commonly employed in quantum semiconductor modeling [10]. The numerical investigations in [11] underline the reasonability of this choice.

The boundaries are assumed to be at grid points 0 and M (where $M + 1$ is the total number of grid points), e.g., at positions $x = 0$ and $x = L$, or in scaled coordinates, $x = 0$ and $x = 1$, respectively. The corresponding boundary conditions are given below:

$$\rho_{k,0} = \sqrt{C_0} \ , \ F_{k,0} = 0 \ , \ V_{k,0} = 0 \ , \tag{3}$$

$$\rho_{k,M} = \sqrt{C_M} \ , \ F_{k,M} = U \ , \ V_{k,M} = U \ , \tag{4}$$

where a new variable $\rho = \sqrt{n}$ has been introduced, and U is an applied voltage.

The initial conditions for ρ, F and V are required to start numerical simulation in equilibrium case ($U = 0$ V). It is natural to set the initial values for F and V to zero, while one has the full freedom to choose/guess the initial value for ρ.

2.3 Discretization in Time and Space

One of the main requirements for the time-discretization is that the scheme should be stable [12,13]. Further, there is no need for higher-order schemes, since the overall discretization error will be dominated by the one introduced through the space-discretization [14,15]. Thus, schemes with first-order accuracy in the time step are sufficient. Moreover, in most of the numerical simulations for the classical Drift-Diffusion Model (DDM), schemes based on backward time differences are employed [12,14]. Since the TQDDM is an $O(\xi^2)$ correction of DDM, it is reasonable to assume this to be true [9]. The most prominent scheme fulfilling the above requirements is the implicit backward Euler method [14,15]. Afterwards, a convenient realizable discrete scheme is derived in two steps: a uniform spatial grid ($\Delta x = L/M$) is introduced, and the finite difference method is used [13,14,15].

3 Geometry and Relevant Parameters

A Quantum Well (QW) is a synthetic heterostructure containing a very thin layer (thickness of a few nanometers) of one semiconductor sandwiched between two (thin) layers of another semiconductor with a larger bandgap [3]. A superlattice is another important nanostructure, which arises if several alternating layers of two materials with different bandgaps are grown one by one. The potential profile of quantum wells and barriers, which show periodicity, is intimately connected with the charge transport properties of the nanostructures [4].

In the present paper, the basic Double Barrier Resonant Tunneling Diode (DBRTD) consists of a quantum well GaAs layer sandwiched between two AlGaAs layers, each 5 nm thick. This resonant structure is itself sandwiched between two spacer GaAs layers of 5 nm thicknesses and supplemented with two contact GaAs regions, each 25 nm thick. The basic n^+-n-n^+ DBRTD in 2D is shown in Fig. 1. A superlattice has a similar structure with more than one barrier-QW-barrier alternating layers sandwiched between the space layers and supported with two contact regions.

The contact regions are highly doped (n^+ type) with $C_m = 10^{24}$ m^{-3}, while the channel is moderately doped (n type) with $C_m = 10^{21}$ m^{-3}. The distribution of charged background ions is described by the doping profile C, which is time independent. Such a device exhibits NDR/C due to electron tunneling through the potential barriers. A typical stationary N-shape IVC is well known from the literature.

The domain in 1D case is the interval $\Omega = [0,L]$, $L > 0$ being the device length. The device length is a sum of all layers/regions of the heterostructure. The relaxation time is fixed at $\tau = 10^{-12}$ s [10]. It is assumed that the devices are operating at liquid-nitrogen temperature, $T_0 = 77$ K. The effective electron mass is chosen to be $m_e^* = 0.067 \cdot m_0$, where m_0 is electron rest mass. The permittivity of GaAs/AlGaAs is chosen as $\varepsilon_s = 13.1 \cdot \varepsilon_0$, where ε_0 is permittivity in vacuum. It is also assumed that we operate close to the thermal equilibrium.

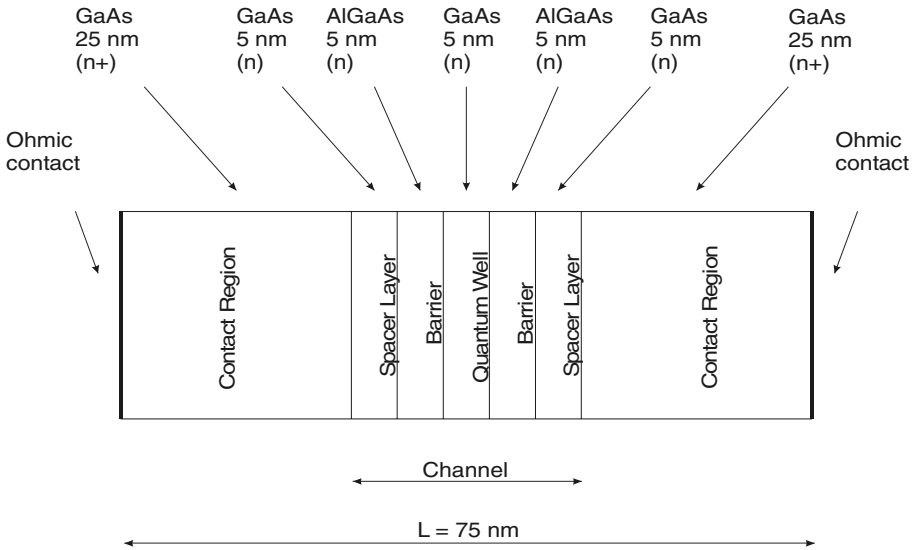


Fig. 1. Basic double barrier resonant tunneling semiconductor structure

4 Numerical Simulation

The numerical simulation is implemented in the Matlab. The system of discretized equations is solved in fully coupled manner for the first time, i.e., in the present work no decoupling algorithms, procedures, or methods are used.

To solve the nonlinear discrete system of equations, which follows from (1), the Newton Iteration Procedure (NIP) is employed [16], where the previous time step is used as initial guess. The following termination criterion is used to stop NIP:

$$\max_{k,l}\left(\left|u_{k,l}^{new}-u_{k,l}^{old}\right|\right) \le \tau_0 10^{-6} \ , \tag{5}$$

where u denotes one of the variables ρ, F and V. The indexes k and l correspond to the time and the space discretization, respectively. Convergence is assumed when the residuals are smaller than a set tolerance.

The scaled time step is changed during the time evolution. Initially, it is set to 10^{-6} and afterwards increased. The algorithm for the time step is based on two criterions: to speed up the time evolution and to ensure the convergence. The maximum number of required steps in the NIP is less than or equal to 5. Stationary solutions are reached after approximately 150 time steps, depending upon the semiconductor heterostructure under consideration. As a test example for the steady state, the total time is fixed to $T = 100\tau$ and the same results are obtained.

A uniform grid is used for space discretization. The resolution is set to 4 ppnm (points per nm). It gives us $M = 300$ for the basic case of a 75 nm DBRTD. We have checked the convergence of calculated variables (ρ, F, and V) by increasing the resolution (decreasing the grid size) and obtained a relative error less than 1% when 4 ppnm is used.

The obtained results for electron density n, quantum quasi-Fermi level F, and electrostatic potential V are smooth and good agreement in the case of a basic DBRTD heterostructure, also considered in [10], is obtained. Excellent symmetry for equilibrium state (U = 0 V) is present for all semiconductor heterostructures under consideration. The current density in steady state has almost the same value at the Ohmic contacts, as required (relative error is less than 0.5%).

5 Results

Here, we present results for different resonant tunneling heterostructures, obtained with the numerical simulation. In the first case, a DBRTD with one QW is considered corresponding to three different QW sizes (2, 5, and 10 nm), while in the second case, superlattices with 2, 3, and 5 QWs are considered, where each QW is of the same size (5 nm). In all presented cases, width of the barriers and space layers are 5 nm. The length of all DBRTDs is fixed to L = 75 nm (contact regions are 26.5, 25, and 22.5 nm, respectively), while the length of the superlattices is a sum of all layers (contact regions are fixed to 25 nm). The external potential U is always applied on the RHS Ohmic contact, while the LHS contact is grounded. In what follows, we are interested in the spatial dependency of electron densities and afterwards we comment on obtained current-voltage (IV) characteristics.

The results for electron densities for all three DBRTD (QW width 2, 5, and 10 nm), are given in Fig. 2. As a reference, doping profiles are also given. The applied potential is U = 0.2 V, which approximately corresponds to the peak voltage for these three cases. The results are qualitatively different for various thicknesses of the QW. For the case of a 2 nm QW, accumulation of electrons in the QW is evident, however, the resulting density for the "peak"-applied voltage is much less than the doping concentration. For the cases of 5 and 10 nm QW, the accumulation effect of electrons in the QW is indeed very important, and electron densities, for the peak-applied voltage, are significantly larger than doping concentration (two orders of magnitude). This indicates that a minimum thickness of the QW exists, which allows sufficient accumulation of electrons. In addition, the electron density inside the QW shows an increase of spatial asymmetry as the thickness of the QW is increased. A significant electron density reduction is apparent in the barrier regions and the minimums are not the same on both sides of the QW. The densities inside barriers have larger values on the side where the external potential is applied, in contrast to the densities inside the QW. The minimum of the density in the barrier changes significantly with increasing thickness of the QW. In contrast, the peak of the electron density inside QWs of differ ent thicknesses shows "saturation", i.e., they are almost the same (if the thickness of the QW is large enough, $L_{QW} \geq$ 4-5 nm).

The results for electron densities for the superlattice heterostructures (with 2, 3, and 5 QWs), are given in Fig. 3. As a reference, doping profiles are also given. The applied potentials are U = 0.30 V, 0.35 V, and 0.50 V, which approximately correspond to the peak voltages, respectively. The superlattice heterostructures have different peaks of electron density in different QWs, under the influence of an external potential. In the case of a superlattice with 2 QWs, the charge accumulation in the QW being closer to the side where the external potential is applied, is much smaller, as is the peak. In the case of a superlattice with more than 2 QWs, the peak of

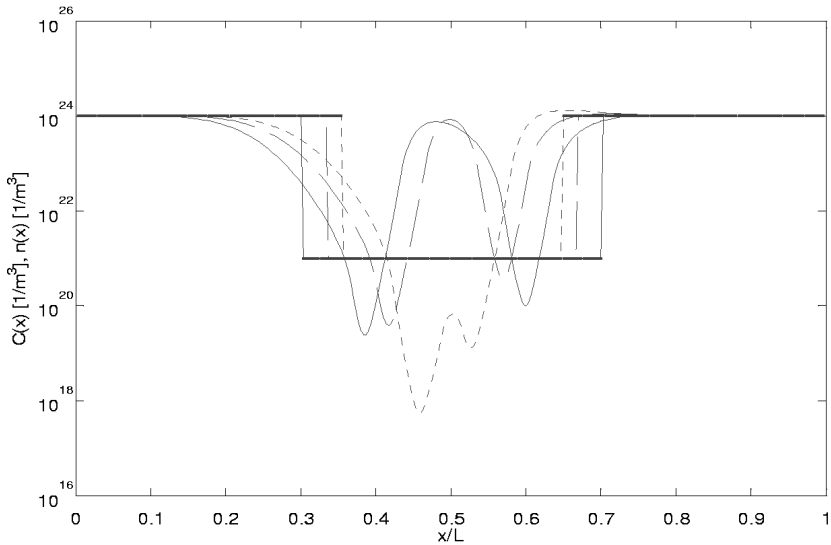


Fig. 2. Densities of electrons $n(x)$ for the DBRTDs with 2, 5, and 10 nm QW; dotted, dashed, and solid curve, respectively; $U = 0.2$ V; corresponding doping profiles $C(x)$ are represented by bold step-curves with the same line coding as for $n(x)$

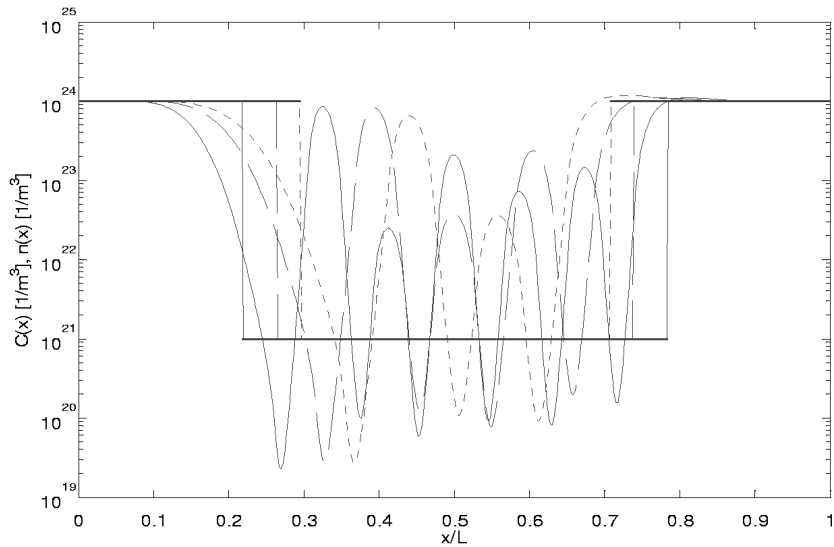


Fig. 3. Densities of electrons $n(x)$ for the superlattices with 2, 3, and 5 QWs; dotted, dashed, and solid curve, respectively; $U = 0.30$ V, 0.35 V, and 0.50 V, respectively; corresponding doping profiles $C(x)$ are represented by bold step-curves with the same line coding as for $n(x)$

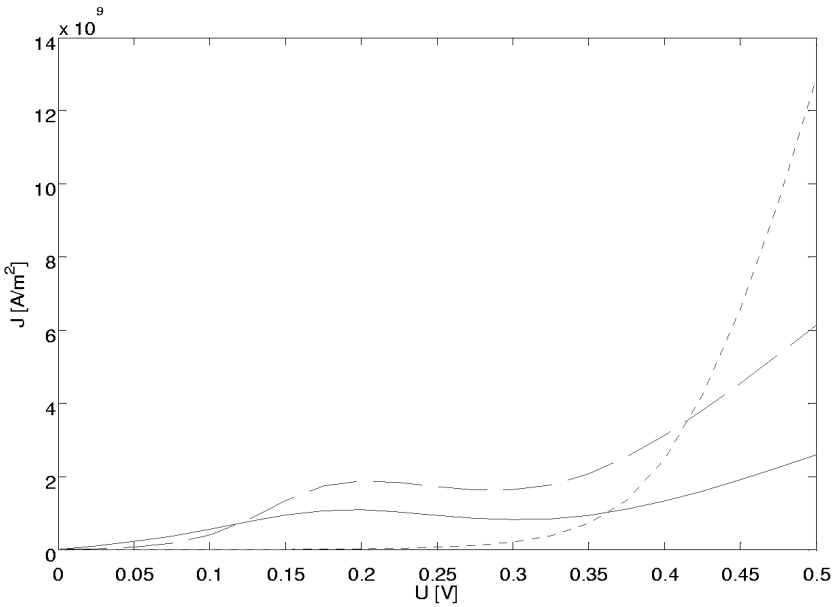


Fig. 4. The IV characteristics for the DBRTDs with 2, 5, and 10 nm QW; dotted, dashed, and solid curve, respectively

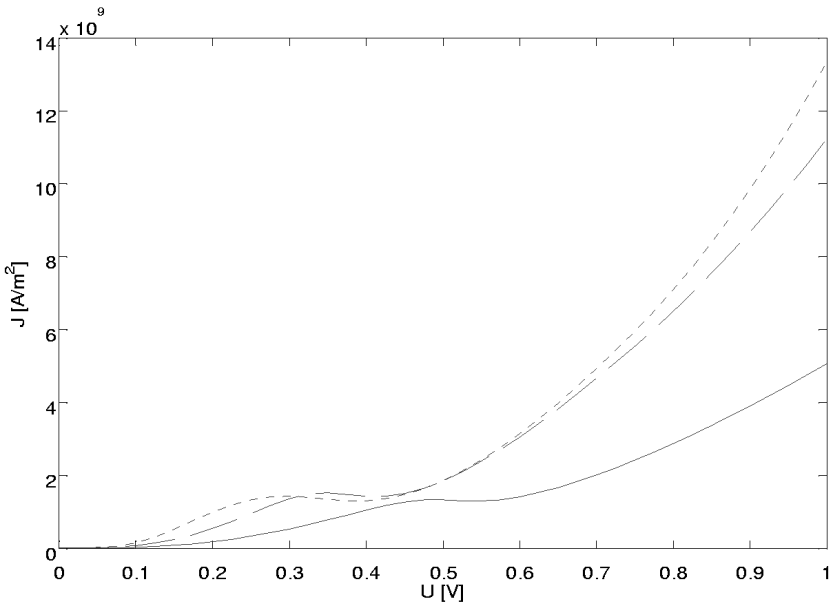


Fig. 5. The IV characteristics for the superlattices with 2, 3, and 5 QWs; dotted, dashed, and solid curve, respectively

the electron density inside QWs changes its magnitude, from the side where the external potential is applied towards the zero-volt (grounded) contact. In general, the QW that is closest to the zero-volt contact has the largest peak of the electron density. In contrast, the minimum of the electron density is always reached in the barrier that is nearby the zero-volt contact.

The IV characteristics for the DBRTDs (QW thickness of 2, 5, and 10 nm), are given in Fig. 4. The IV characteristic for the DBRTD with 2 nm QW does not show NDR/C at all. Changing the width of the QW, a significant change in the IV characteristic occurs, i.e., the current density for the same applied voltage is changed. In addition, the peak and the valley current density, and the relative ratio between them, also change significantly. However, the peak and the valley potential will only slightly differ.

The IV characteristics for the superlattice heterostructures with 2, 3, and 5 QWs (corresponding to L = 85, 95, 115 nm, respectively), are given in Fig. 5. It is obvious that an increase of the number of QWs leads to a significant influence on the IV characteristic. At the same time, the peak and the valley potential are also changed, and the current density is reduced. However, the peak and the valley current density will be only slightly changed.

Unfortunately, it is extremely difficult to predict the exact values of the peak and the valley current density in the N-shaped IV characteristics. These values are strongly affected by other mechanisms, such as phonon-assisted tunneling, impurity-assisted tunneling, and scattering. The voltage positions of the current density peaks and valleys are, however, easier to establish, since they are related to energy levels of the subbands.

The computed values of the current density strongly depend on the choice of the intrinsic device parameters. The most important parameters are the effective electron mass and the relaxation time. Thus, the choice of the intrinsic parameters is crucial for the accurate quantitative simulation of the resonant tunneling heterostructures. However, we expect the general tendencies observed here to be correct.

6 Conclusion

The system of fully discretized coupled nonlinear algebraic equations, which follows from (1), is possible to solve avoiding the use of decoupling algorithms, procedures, or methods. The results obtained for different resonant tunneling heterostructures using TQDDM show that the electron density is strongly nonlinear and asymmetric in both, QWs and barriers, under the influence of an external potential. However, in the equilibrium case (U = 0 V), perfect symmetry is present. The IV characteristics of the heterostructure nanodevices are quite different, when varying the thickness and the number of QWs of the device. In particular, changing the thickness of the QW, the peak and the valley current density change significantly, while the peak and the valley potential only slightly differ. In contrast, changing the number of the QWs (keeping the size of the QW constant), the peak and the valley potential change dramatically, while the peak and the valley current are almost unchanged.

References

1. Sze, S.M.: Semiconductor Devices – Physics and Technology. Wiley, New York (1985)
2. Shur, M.: Physics of Semiconductor Devices. Prentice Hall, Englewood Cliffs (1990)
3. Yu, P.Y., Cardona, M.: Fundamentals of Semiconductors – Physics and Materials Properties. Springer-Verlag, Berlin Heidelberg New York (1996)
4. Scholl, E.: Nonlinear Spatio-Temporal Dynamics and Chaos in Semiconductors. Cambridge University Press, Cambridge (2001)
5. Ancona, M.G.: Diffusion-Drift Modeling of Strong Inversion Layers. COMPEL, 6 (1987) 11–18
6. Ancona, M.G., Tiersten, H.F.: Macroscopic Physics of the Silicon Inversion Layer. Phys. Rev. B, Vol. 35, No. 15 (1987) 7959–7965
7. Ancona, M.G., Iafrate, G.J.: Quantum Correction to the Equation of State of an Electron Gas in a Semiconductor. Phys. Rev. B, Vol. 39, No. 13 (1989) 9536–9540
8. Pinnau, R., Unterreiter, A.: The Stationary Current-Voltage Characteristics of the Quantum Drift-Diffusion Model. SIAM J. Numer. Anal., Vol. 37, No. 1 (1999) 211–245
9. Pinnau, R.: The Linearized Transient Quantum Drift-diffusion Model – Stability of Stationary States. ZAMM, 80(5) (2000) 327–344
10. Pinnau, R.: Numerical Approximation of the Transient Quantum Drift-Diffusion Model. Nonlinear Analysis, Vol. 47 (2001) 5849–5860
11. Pinnau, R.: A Note on Boundary Conditions for Quantum Hydrodynamic Equations. Appl. Math. Lett., 12 (1999) 77–82
12. Markowich, P.A., Ringhofer, C.A.: Stability of the Linearized Transient Semiconductor Device Equations. ZAMM, 67(7) (1987) 319–332
13. Markowich, P.A., Ringhofer, C.A., Schmeiser, C.: Semiconductor Equations. Springer-Verlag, Wien (1991)
14. Mock, M.S.: Analysis of Mathematical Models of Semiconductor Devices. Boole Press, Dublin (1983)
15. Selberherr, S.: Analysis and Simulation of Semiconductor Devices. Springer-Verlag, Wien New York (1984)
16. Schatzman, M.: Numerical Analysis – A Mathematical Introduction. Clarendon Press, Oxford (2002)

Teletraffic Generation of Self-Similar Processes with Arbitrary Marginal Distributions for Simulation: Analysis of Hurst Parameters

Hae-Duck J. Jeong[1], Jong-Suk R. Lee[2], and Hyoung-Woo Park[2]

[1] Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
hdj11@ext.canterbury.ac.nz
[2] Grid Technology Research Department, Supercomputing Centre
Korea Institute of Science and Technology Information
Daejeon, Korea
jsruthlee@kisti.re.kr

Abstract. Simulation studies of telecommunication networks require a mechanism to transform self-similar processes into processes with arbitrary marginal distributions. The problem of generating a self-similar process of a given marginal distribution and an autocorrelation structure is difficult and has not been fully solved. Our results presented in this paper provide clear experimental evidence that the long-range dependent (LRD) self-similarity of the input process is not preserved in the output process generated by the inverse cumulative distribution function (ICDF) transformation, if the output process has an infinite variance. On the basis of our results we formulate the following hypothesis:
If the ICDF transformation is applied to LRD self-similar processes with normal marginal distributions, then it preserves H parameter of the input process if the output marginal distribution has a finite variance.

1 Introduction

Simulation studies of telecommunication networks often require generation of random variables, or stochastic processes, characterised by different probability distributions. Thus far we have discussed generation of self-similar sequences with a normal marginal distribution. We can obtain sequences of numbers from normal distributions with different mean values and variances by applying such standard transformations as shifting and rescaling/normalisation. In practical simulation studies, however, generation of self-similar processes of several different non-normal marginal probability distributions might be required. The most common method of transforming realisations of one random variable into realisations of another random variable is based on the inverse of cumulative distribution functions. This method and its application in transformations of self-similar processes are discussed in Section 2 in detail.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 827–836, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The theory of transformations of strictly and second-order self-similar processes has not been fully developed. In this paper, we look at applications of the inverse cumulative distribution function (ICDF) transformation[1] to the generation of long-range dependent (LRD) sequences governed by non-normal marginal distributions from LRD sequences of normal marginal distributions.

For studying the properties of the ICDF transformation in the context of self-similar process we investigate its properties when it is applied to the exact self-similar process, taking the self-similar fractional Gaussian noise (FGN) process as the reference [5], [6], [7], [8], [9]. This FGN process was generated by the *Durbin-Levinson* algorithm, described in [5], [10]. We consider output processes with different marginal probability distributions (exponential, gamma, Pareto, uniform and Weibull), with finite and infinite variances, and compare H parameters of output processes with those characterising input self-similar FGN processes. Our findings are summarised in Section 4.

2 Generation of LRD Self-Similar Processes with Arbitrary Marginal Distributions

Simulation studies of telecommunication networks require a mechanism to transform self-similar processes into processes with arbitrary marginal distributions [9], [11], [12]. In this paper, we investigate preservation of the LRD self-similarity in output processes with different marginal distributions when transforming exact self-similar FGN processes into processes with five different marginal distributions (exponential, gamma, Pareto, uniform and Weibull), with finite and infinite variances, using the ICDF transformation.

2.1 The Methods of the ICDF Transformation

The ICDF transformation is based on the observation that given any random variable X_i with a cumulative distribution function (CDF) $F(x)$, the random variable $u = F(x)$ is independent and uniformly distributed between 0 and 1. Therefore, x can be obtained by generating uniform realisations and calculating $x = F^{-1}(u)$.

We assume that a process $\mathbf{X}$ is a Gaussian process with zero mean, variance of one and a given autocorrelation function (ACF) $\{\rho_k\}$. Let $F_X(x)$ be its marginal CDF and $F_Y(y)$ be a marginal CDF of the process $\mathbf{Y}$. The process $\mathbf{Y}$ with the desired marginal CDF $F_Y(y)$ can be generated by the ICDF transformation from the process $\mathbf{X}$. Following the ICDF transformation, when transforming a random variable X_i into a random variable Y_i, we use the formula:

$$F_X(x) = F_Y(y), \tag{1}$$

[1] The TES (Transform-Expand-Sample) process [1], [2] and the ARTA (Autoregressive-to-Anything) process [3], [4] could be used the generation of correlated sequences.

Thus:

$$y = F_Y^{-1}\left(F_X(x)\right) \tag{2}$$

hence the method is called the ICDF transformation.

Here we consider five marginal distributions of output processes: exponential, gamma, Pareto, uniform and Weibull distributions that are frequently used in simulation practice.

Exponential Marginal Probability Distribution: The exponential distribution has the CDF

$$F_Y(y) = \begin{cases} 0, & \text{for } \ y \leq 0, \\ 1 - e^{-\lambda y}, & \text{for } \ y > 0, \end{cases} \tag{3}$$

where λ is the mean of a random variable Y.

To generate a random variable Y with an exponentially distribution from a random variable X of normal distribution, one applies the transformation:

$$y_i = -(\frac{1}{\lambda}) * \log(F_X(x_i)), \tag{4}$$

where $F_X(\cdot)$ is the CDF of the normal distribution.

Gamma Marginal Probability Distribution: The gamma distribution has the CDF

$$F_Y(y) = \begin{cases} 0, & \text{for } \ y \leq 0, \\ 1 - e^{-y/\beta_\Gamma} \sum_{j=0}^{\alpha_\Gamma - 1} \frac{(y/\beta_\Gamma)^j}{j!}, & \text{for } \ y > 0, \end{cases} \tag{5}$$

if α_Γ (the shape parameter) is a natural number, and β_Γ is the scale parameter, $\beta_\Gamma > 0$. If α_Γ is not integer, then there is no closed form of the CDF for the gamma distribution. A few methods for generating pseudo-random numbers governed by such a gamma probability distribution have been proposed [13] (pp. 487-490). We chose the Newton-Raphson technique, and used an implementation of this technique given in [14].

Pareto Marginal Probability Distribution: The Pareto distribution has the CDF

$$F_Y(y) = \begin{cases} 0, & \text{for } \ y < 1, \\ 1 - \left(\frac{b}{y}\right)^\alpha, & \text{for } \ 1 \leq y \leq \infty, \end{cases} \tag{6}$$

where α is a shape parameter and b is the minimum allowed value of y, $0 < b \leq y$. We assume $b = 1$. To generate random variables with a Pareto distribution Y from random variables of normal distribution X, one applies the transformation:

$$y_i = 1/(F_X(x_i))^{1/\alpha}. \tag{7}$$

Uniform Marginal Probability Distribution: The uniform distribution has the CDF

$$F_Y(y) = \begin{cases} 0, & \text{for } y < a, \text{and } b < y \\ \frac{y-a}{b-a}, & \text{for } a \le y \le b, \end{cases} \tag{8}$$

where a is a lower limit and b is an upper limit, $a < b$. To generate pseudo-random numbers with a uniform distribution Y from random variables of normal distribution X, one applies the transformation:

$$y_i = a + (b - a)F_X(x_i). \tag{9}$$

Weibull Marginal Probability Distribution: The Weibull distribution has the CDF

$$F_Y(y) = \begin{cases} 0, & \text{for } y \le 0, \\ 1 - e^{-(y/\beta)^\alpha}, & \text{for } 0 < y, \end{cases} \tag{10}$$

where α is a shape parameter and β is a scale parameter. To generate a random variable with a Weibull distribution Y from a random variable of normal distribution X, one applies the transformation:

$$y_i = \beta \left(-\log(F_X(x_i))\right)^{1/\alpha}. \tag{11}$$

2.2 Effects of Transformation

In simulation studies of such stochastic dynamic processes as those that occur in telecommunication networks one needs to decide both about their marginal probability distributions and autocorrelation structures. The problem of generating a strictly and/or second-order self-similar process of a given marginal distribution and an autocorrelation structure is difficult and has not been fully solved. No existing procedure is entirely satisfactory in terms of mathematical rigour, computational efficiency, accuracy of approximation, and precise and concise parameterisation [15].

Applications of the transformation in Equation (2) to transformations of correlated processes have been studied by several researchers [1], [5], [7], [16]. In general, as proved by Beran (see [17], pp. 67-73), a transformation $y = G(x)$ applied to a strictly and/or second-order LRD self-similar sequence of numbers $\{x_1, x_2, \ldots\}$ does not preserve LRD properties in the output sequence $\{y_1, y_2, \ldots\}$. However, as proved in [16], if in Equation (2):

1. $F_X(\cdot)$ represents normal distribution,
2. $\{x_1, x_2, \ldots\}$ is an LRD self-similar sequence,
3. the transformation $G^2(x)$ is integrable, i.e.,

$$\int_{-\infty}^{+\infty} G^2(x) dF_X(x) < \infty, \quad \text{and} \tag{12}$$

4. $E(XY) \neq 0$,

then the output sequence $\{y_1, y_2, \ldots\}$ is asymptotically self-similar, with the same coefficient H as the sequence $\{x_1, x_2, \ldots\}$.

Related issues have been investigated. Wise et al. [18] and Liu and Munson [19] showed that, following the transformation of marginal distribution, the transformation of ordinary ACF can be characterised when the input process is normal. They also indicated other processes for which this could be applied. Huang et al. [16] demonstrated that, if the process $\mathbf{X}$ is self-similar and has a normal marginal distribution, under general conditions, the output process $\mathbf{Y}$ is an asymptotically self-similar process with the same Hurst parameter ($\frac{1}{2} < H < 1$); for proof of the invariance of the Hurst parameter H, see [16]. Geist and Westall [20] demonstrated that arrival processes, obtained by the FFT method proposed by Paxson [9], have ACFs that are consistent with LRD. However, it has not been fully developed to generate self-similar processes with arbitrary marginal distributions from self-similar processes with (normal) marginal distributions and autocorrelation structures [15], [20].

3 Numerical Results

The numerical results of this section are used to investigate how well the LRD self-similarity of the original Gaussian processes are preserved when they are converted into processes with non-normal marginal distributions. For each of H = 0.6, 0.7, 0.8 and 0.9, 100 exact self-similar sample sequences of 32,768 (2^{15}) numbers starting from different random seeds are used.

The following five different marginal distributions are investigated: the exponential distribution with $\lambda = 9$; the uniform distribution with $a = 0$ and b = 1; the gamma distribution with $\alpha = 2$ and $\beta = 1$; the Pareto distributions with α = 1.2, 1.4, 1.6, 1.8 and 20.0; and the Weibull distribution with $\alpha = 2$ and β = 1.

3.1 Analysis of H Parameters

For more rigorous proof, we analyse the self-similar sequences with five different marginal distributions generated by the exact self-similar FGN process using the wavelet-based H estimator and Whittle's MLE[2]. Table 1 shows the estimated mean $\hat{H}$ values of the resulting process. For H = 0.6, 0.7, 0.8 and 0.9, each mean $\hat{H}$ value is obtained from 100 replications. We give 95% confidence intervals for the means in parentheses. All results in Tables 1 – 2 are presented together with their relative errors ΔH, defined as:

$$\Delta H = \frac{\hat{H} - H}{H} * 100\%, \tag{13}$$

[2] Two estimators are used to analyse H parameters because our results have shown that the wavelet-based H estimator and Whittle's MLE are the least biased of the H estimation techniques. For detailed discussions, see [5].

Table 1. Relative inaccuracy, ΔH, of mean values of estimated H obtained using the wavelet-based H estimator for the exact self-similar FGN process with different marginal distributions for $H = 0.6$, 0.7, 0.8 and 0.9. We give 95% confidence intervals for the mean values in parentheses.

Distribution	Mean Values of Estimated H and ΔH							
	.6		.7		.8		.9	
	$\hat{H}$	$\Delta H(\%)$	$\hat{H}$	$\Delta H(\%)$	$\hat{H}$	$\Delta H(\%)$	$\hat{H}$	$\Delta H(\%)$
Exponential	.5879 (.560,.615)	-2.039	.6830 (.656,.711)	-2.521	.7800 (.753,.808)	-2.604	.8797 (.852,.907)	-2.356
Gamma	.5945 (.567,.622)	-0.949	.6922 (.665,.720)	-1.198	.7909 (.763,.818)	-1.243	.8912 (.864,.919)	-1.079
Uniform	.5971 (.570,.625)	-0.514	.6964 (.669,.724)	-0.604	.7953 (.768,.823)	-0.700	.8929 (.865,.920)	-0.892
Weibull	.5981 (.571,.626)	-0.348	.6979 (.670,.725)	-0.394	.7976 (.770,.825)	-0.410	.8975 (.870,.925)	-0.387
Pareto ($\alpha = 20.0$)	.5857 (.558,.613)	-2.378	.6800 (.653,.708)	-2.862	.7765 (.749,.804)	-2.940	.8764 (.849,.904)	-2.626
Pareto ($\alpha = 1.2$)	.5014 (.474,.529)	-16.43	.5027 (.475,.530)	-28.19	.5333 (.506,.561)	-33.34	.6300 (.603,.658)	-30.00
Pareto ($\alpha = 1.4$)	.5098 (.482,.537)	-15.03	.5237 (.496,.551)	-25.18	.5690 (.542,.597)	-28.87	.6778 (.650,.705)	-24.69
Pareto ($\alpha = 1.6$)	.5189 (.491,.546)	-13.52	.5468 (.519,.574)	-21.89	.6047 (.577,.632)	-24.42	.7177 (.690,.745)	-20.25
Pareto ($\alpha = 1.8$)	.5281 (.501,.556)	-11.99	.5690 (.542,.597)	-18.71	.6362 (.609,.664)	-20.47	.7495 (.722,.777)	-16.73

where H is the exact value and $\hat{H}$ is the mean of the estimates. These results in Table 1 show that all confidence intervals are within the required values, except for those with Pareto distribution is $\alpha = 1.2$, 1.4, 1.6 and 1.8. Values in Table 2 show that, for gamma ($H = 0.6$), uniform ($H = 0.6$ and 0.7) and Weibull ($H = 0.6$, 0.7, 0.8 and 0.9), confidence intervals are within the required values, but others are slightly underestimated (i.e., $|\Delta H| < 4\%$).

If one considers output marginal distributions with infinite variances, then as it was proved in [16], the H values of the input process are not preserved. This fact is illustrated by results presented in Tables 1 – 2, where Pareto distributions with infinite variances ($\alpha = 1.2$, 1.4, 1.6 and 1.8) have been added to the previously considered five output distributions with finite variances for $H = 0.6$, 0.7, 0.8 and 0.9.

On the basis of our results we formulate the following hypothesis:

If transformation (2) is applied to LRD self-similar processes with normal marginal distributions, then it preserves H parameter if the output marginal distribution has a finite variance.

We think that this hypothesis could be analytically proved by showing that in the case of an infinite variance, the transformation in Equation (2) does not satisfy the assumption that its squared form in Equation (12) must be integrable [16].

Table 2. Relative inaccuracy, ΔH, of mean values of estimated H obtained using Whittle's MLE for the exact self-similar FGN process with different marginal distributions for $H = 0.6$, 0.7, 0.8 and 0.9. We give 95% confidence intervals for the mean values in parentheses.

Distribution	Mean Values of Estimated H and ΔH							
	.6		.7		.8		.9	
	$\hat{H}$	$\Delta H(\%)$	$\hat{H}$	$\Delta H(\%)$	$\hat{H}$	$\Delta H(\%)$	$\hat{H}$	$\Delta H(\%)$
Exponential	.5856 (.576,.595)	-2.394	.6774 (.668,.687)	-3.230	.7749 (.766,.784)	-3.134	.8797 (.871,.889)	-2.258
Gamma	.5923 (.583,.602)	-1.290	.6878 (.678,.697)	-1.748	.7864 (.777,.796)	-1.701	.8892 (.880,.898)	-1.202
Uniform	.5962 (.587,.606)	-0.628	.6928 (.683,.702)	-1.034	.7897 (.781,.799)	-1.285	.8889 (.880,.898)	-1.235
Weibull	.5981 (.589,.608)	-0.313	.6969 (.688,.706)	-0.445	.7964 (.787,.806)	-0.451	.897 (.888,.906)	-0.335
Pareto ($\alpha = 20.0$)	.5833 (.574,.593)	-2.780	.6737 (.664,.683)	-3.760	.7706 (.761,.780)	-3.678	.8759 (.867,.885)	-2.674
Pareto ($\alpha = 1.2$)	.5050 (.495,.515)	-15.84	.5155 (.506,.525)	-26.35	.5438 (.534,.554)	-32.03	.6393 (.630,.649)	-28.96
Pareto ($\alpha = 1.4$)	.5091 (.499,.519)	-15.15	.5277 (.518,.537)	-24.61	.5666 (.557,.576)	-29.18	.6774 (.668,.687)	-24.73
Pareto ($\alpha = 1.6$)	.5152 (.506,.525)	-14.13	.5426 (.533,.552)	-22.49	.5919 (.582,.602)	-26.01	.7112 (.702,.721)	-20.97
Pareto ($\alpha = 1.8$)	.5224 (.513,.532)	-12.93	.5584 (.549,.568)	-20.23	.6169 (.607,.626)	-22.89	.7399 (.731,.749)	-17.79

3.2 Variances for Estimated H

Tables 3 – 4 show variances for estimated H obtained using the wavelet-based H estimator and Whittle's MLE for the exact self-similar FGN process with different marginal distributions for $H = 0.6$, 0.7, 0.8 and 0.9. Estimated variances for the output processes with five different marginal distributions were slightly higher than the original, but those with the Pareto marginal distribution with $\alpha = 1.2$, 1.4, 1.6 and 1.8 had the highest variances. All variances gradually increased as the H value increased.

4 Conclusions

We investigated how well the LRD self-similarity of the original processes were preserved when the self-similar processes were converted into suitable self-similar processes with five exponential, gamma, Pareto, uniform and Weibull marginal distributions. We used the ICDF transformation to produce self-similar processes with five different marginal distributions for the stochastic simulation of telecommunication networks with self-similar teletraffic. Our results presented in this paper provide clear experimental evidence that the LRD self-similarity of the input process is not preserved in the output process generated by trans-

Table 3. Variances for estimated H obtained using the wavelet-based H estimator for self-similar processes with different marginal distributions for $H = 0.6$, 0.7, 0.8 and 0.9.

Distribution	Variances of Estimated H			
	.6	.7	.8	.9
Exponential	1.6620e-04	2.0330e-04	2.8780e-04	4.5280e-04
Gamma	1.9940e-04	2.0250e-04	2.1560e-04	2.6410e-04
Uniform	1.9930e-04	1.9920e-04	1.9620e-04	2.1290e-04
Weibull	1.8120e-04	1.9100e-04	2.0760e-04	2.3660e-04
Pareto ($\alpha = 20.0$)	1.6890e-04	2.0980e-04	3.0880e-04	4.9820e-04
Pareto ($\alpha = 1.2$)	5.0160e-03	1.0020e-02	9.7020e-03	9.1550e-03
Pareto ($\alpha = 1.4$)	3.5150e-03	6.6330e-03	7.1350e-03	7.3760e-03
Pareto ($\alpha = 1.6$)	2.4050e-03	4.4900e-03	5.5260e-03	5.7900e-03
Pareto ($\alpha = 1.8$)	1.6220e-03	3.0460e-03	4.2630e-03	4.3950e-03

Table 4. Variances for estimated H obtained using Whittle's MLE for self-similar processes with different marginal distributions for $H = 0.6$, 0.7, 0.8 and 0.9.

Distribution	Variances of Estimated H			
	.6	.7	.8	.9
Exponential	1.2697e-05	1.5443e-05	2.0052e-05	3.0836e-05
Gamma	1.1583e-05	1.2920e-05	1.4497e-05	1.6641e-05
Uniform	1.1518e-05	1.2855e-05	1.4325e-05	1.7971e-05
Weibull	1.1581e-05	1.2447e-05	1.3394e-05	1.5497e-05
Pareto ($\alpha = 20.0$)	1.3430e-05	1.7630e-05	2.3820e-05	4.0410e-05
Pareto ($\alpha = 1.2$)	1.0190e-04	2.5820e-04	1.0370e-03	5.0670e-03
Pareto ($\alpha = 1.4$)	9.6520e-05	3.6780e-04	1.3100e-03	4.9850e-03
Pareto ($\alpha = 1.6$)	1.0100e-04	4.6260e-04	1.5050e-03	4.4040e-03
Pareto ($\alpha = 1.8$)	1.0280e-04	5.0840e-04	1.5720e-03	3.6650e-03

formation (2), if the output process has an infinite variance. On the basis of our results we formulate the following hypothesis:

> *If transformation (2) is applied to LRD self-similar processes with normal marginal distributions, then it preserves H parameter of the input process if the output marginal distribution has a finite variance.*

Further research work is needed to investigate exact values of preservation of the second-order LRD self-similarity when transforming second-order self-similar processes into processes with arbitrary marginal distributions.

Acknowledgements. The authors acknowledge Dr. Manfred Jobmann, Dr. Don McNickle and Dr. Krzysztof Pawlikowski for their valuable comments. The authors also wish to thank the financial support of Korea Institute of Science and Technology Information, Korea.

References

1. Melamed, B.: TES: a Class of Methods for Generating Autocorrelated Uniform Variates. ORSA Journal on Computing **3** (1991) 317–329
2. Melamed, B., Hill, J.R.: A Survey of TES Modeling Applications. Simulation (1995) 353–370
3. Cario, M., Nelson, B.: Autoregressive to Anything: Time-Series Input Processes for Simulation. Operations Research Letters **19** (1996) 51–58
4. Cario, M., Nelson, B.: Numerical Methods for Fitting and Simulating Autoregressive-to-Anything Processes. INFORMS Journal on Computing **10** (1998) 72–81
5. Jeong, H.D.J.: Modelling of Self-Similar Teletraffic for Simulation. PhD thesis, Department of Computer Science, University of Canterbury (2002)
6. Jeong, H.D.J., McNickle, D., Pawlikowski, K.: Generation of Self-Similar Time Series for Simulation Studies of Telecommunication Networks. In: Proceedings of the First Western Pacific and Third Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management, Christchurch, New Zealand (1999) 221–230
7. Jeong, H.D.J., McNickle, D., Pawlikowski, K.: Generation of Self-Similar Processes for Simulation Studies of Telecommunication Networks. Mathematical and Computer Modelling **38** (2003) 1249–1257
8. Neame, T.: Characterisation and Modelling of Internet Traffic Streams. PhD thesis, Department of Electrical and Electronic Engineering, The University of Melbourne (2003)
9. Paxson, V.: Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic. Computer Communication Review, ACM SIGCOMM **27** (1997) 5–18
10. Abry, P., Flandrin, P., Taqqu, M., D.Veitch: Self-Similarity and Long-Range Dependence Through the Wavelet Lens. In: Theory and Applications of Long-Range Dependence. Birkhäuser, Doukhan, Oppenheim, and Taqqu (eds), Boston, MA (2002) 527–556
11. Leroux, H., Hassan, M.: Generating Packet Inter-Arrival Times for FGN Arrival Processes. In: The 3rd New Zealand ATM and Broadband Workshop, Hamilton, New Zealand (1999) 1–10
12. Leroux, H., Hassan, M., Egudo, R.: On the Self-Similarity of Packet Inter-Arrival Times of Internet Traffic. In: The 3rd New Zealand ATM and Broadband Workshop, Hamilton, New Zealand (1999) 11–19
13. Law, A., Kelton, W.: Simulation Modeling and Analysis. 2nd ed., McGraw-Hill, Inc., Singapore (1991)
14. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C. Cambridge University Press, Cambridge (1999)
15. Geist, R., Westall, J.: Practical Aspects of Simulating Systems Having Arrival Processes with Long-Range Dependence. In: Proceedings of the 2000 Winter Simulation Conference, Orlando, Florida, USA, J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick (eds.) (2000) 666–674
16. Huang, C., Devetsikiotis, M., Lambadaris, I., Kaye, A.: Modeling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach. Computer Communication Review, Proceedings of ACM SIGCOMM'95 **25** (1995) 114–125
17. Beran, J.: Statistics for Long-Memory Processes. Chapman and Hall, New York (1994)

18. Wise, G., Traganitis, A., Thomas, J.: The Effect of a Memoryless Nonlinearity on the Spectrum of a Random Process. IEEE Transactions on Information Theory **IT-23** (1977) 84–89
19. Liu, B., Munson, D.: Generation of a Random Sequence Having a Jointly Specified Marginal Distribution and Autocovariance. IEEE Transactions on Acoustics, Speech and Signal Processing **ASSP-30** (1982) 973–983
20. Geist, R., Westall, J.: Correlational and Distributional Effects in Network Traffic Models. Performance Evaluation **44** (2001) 121–138

Design, Analysis, and Optimization of LCD Backlight Unit Using Ray Tracing Simulation

Joonsoo Choi[1], Kwang-Soo Hahn[1], Heekyung Seo[1], and Seong-Cheol Kim[2]

[1] School of Computer Science, Kookmin University, Republic of Korea
{jschoi,kshahn}@kookmin.ac.kr
{hkseo}@cs-mail.kookmin.ac.kr
[2] School of Electrical Engineering and Computer Science,
Seoul National University, Republic of Korea
sckim@maxwell.snu.ac.kr

Abstract. The design of BLU for LCD devices, whose goal is to achieve uniform illumination and high luminance across the LCD surface, requries an assistance of illumination design programs. The goal of this paper is to develop a design and analysis tool to model an efficient BLU. The rendering techniques traditionally used in the field of computer graphics are the usual tools of choice to analyze BLU. An analysis method based on Monte Carlo photon tracing to evaluate the optical performance of BLU is presented. An optimization technique based on direct search method, a simplex method by Nelder and Mead, to achieve an optimal uniform illumination is also discussed.

1 Introduction

A liquid crystal display (LCD) is a standard display device for hand-held systems such as notebook computer, PDA, cellular phone, etc. Since liquid crystals are not light-emitting materials, backlight unit (BLU) which is usually placed behind the LCD panel is used for an LCD system as a light source. A typical BLU consists of a light-guiding plate (LGP) and a light source which is located at edges of the LGP to minimize the thickness of the unit. LGP is an optically transparent plate which is rectangular or wedge in shape. Radiated light from the source is conducted into the LGP and is guided inside the LGP based on the principle of total internal reflection. The light is reflected by an array of diffusive ink dots and emitted out the front face of the LGP. The emanated light is dispersed by the diffusing sheet and collimated by prism sheets before it eventually reaches the viewer's eye.

The design of BLU, whose goal is to maximize the light intensity and control the light distribution on the front face of LGP, requires the assistance of illumination design programs. The rendering techniques traditionally used in the field of computer graphics are the usual tools of choice to analyze BLU [1, 2]. We describe an analysis method based on Monte Carlo photon tracing [3, 4] to evaluate the optical performance of BLU.

One of the design challenges of BLU is to achieve a proper uniformity of the emanated light on the surface of LGP. To achieve the uniformity, the arrangement and a density or fill factor gradation of the diffusing ink dots are controlled. Usually the

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 837–846, 2004.
© Springer-Verlag Berlin Heidelberg 2004

diameters of ink dots increase along the propagation direction of light in the LGP. It is difficult to control the density of ink dots by manual operation so that it contributes to the generation of illumination uniformity on the front face of BLU. Therefore optimization technique to automatically compute the best values for the variable parameters is needed to achieve an optimal uniform illumination. The optimization in the field of designing optical illumination devices is an immature field and it is hardly to find a design tool that implements a general optimization algorithm [5].

The objective function in the optimization problem to achieve uniform illumination has some characteristics that are unattractive to apply standard powerful optimization techniques like Newton-based and quasi-Newton methods [6]. Furthermore computation of the objective function is very expensive and time-consuming when the number of photons generated to simulate the performance of BLU is very large. Therefore it is desirable to find an optimization method locating a minimizer in as few function evaluations as possible. Direct search method is a potential candidate for the optimization since it uses only function values and does not require a gradient. The simplex method devised by Nelder and Mead [7] which is widely used method in the class of direct search method is selected to implement the optimization. In this paper, we discuss a method to compute an optimal density function that controls the diameter of ink dots by the simplex method to generate a uniform luminance on the front face BLU.

2 Structure of BLU

A conventional BLU for LCD display in general use for notebook computers has a structure similar to that shown in Figure 1. In this type of BLU, cold-cathode fluorescent lamp (CCFL) is used as the primary light source. The light source is located on one edge of the module and a light guide plate (LGP) is installed for light to travel from the source to the viewer.

The radiated light from the light sources are guided into a LGP by a lamp reflector. LGP is a device designed to transport light from the light source to a point at some distance with minimal loss. Light is propagated along the length of an LGP by means of total internal reflection. LGP is an optically transparent substrate usually made of polymethyl methacrylate (PMMA) and is rectangular or wedge in shape. The primary role of LGP is to extract the light in a direction perpendicualr to the direction of propagation, i.e., to a front surface of an LGP.

In conventional LGP, diffusing ink spots are printed on the back surface of the LGP. A portion of the guided light incident on the diffusing ink is reflected toward the front surface of the LGP. At the same time, small portion of the light rays that do not satisfy the condition of total internal reflection leaks out to the back and side surface of the LGP. To reuse the light by bringing back into the LGP, a reflective sheet is pasted on the back and side surface of the LGP . The emanated light over the LGP by dispersion from the diffusive ink is spreaded uniformly using a light-diffusing sheet, so that the viewer would not see the ink pattern on the back surface of the LGP. Two distinct prism sheets are used to collimate the light spreaded by the diffusing sheet into the direction perpendcular to the front face of an LGP, and therefore improve normal luminance.

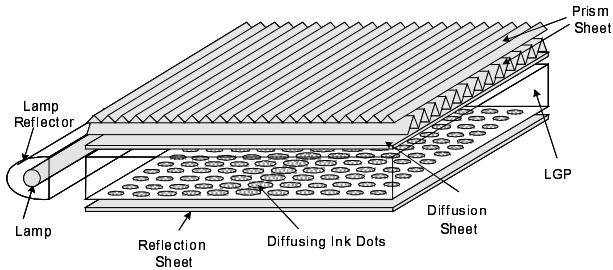


Fig. 1. A structure of conventional BLU. The lamp on the right edge of LGP is missing in this figure. The diameter of scattering ink spot increases along the light propagation direction in the LGP

To achieve a proper illumination uniformity by the emanated light on the surface of the LGP, a fill factor gradation is applied to control the size of each ink spot in an arrangement of ink spots. The size of ink spot positioned further away from the light source is relatively larger so that significant portion of the light with weak intensity because of its long propagation distance is reflected by the large ink spots.

3 Ananysis of BLU with Monte Carlo Photon Tracing

The design of BLU, whose goal is to maximize the luminous intensity and perfectly control the light distribution on the front face of LGP, requires the assistance of illumination design programs. The rendering techniques traditionally used in the field of computer graphics to generate synthetic images are the usual tools of choice to analyze illumination devices, like BLU. We describe an analysis method based on Monte Carlo photon tracing [3, 4] to evaluate the optical performance of BLU. The MC photon tracing simulates illumination by recursive stochastic sampling of illumination in the environment, starting with photons from the light sources.

Each photon is traced along its straight trajectory until the photon hits the nearest surface. At the photon-surface intersection position, the photon is to be reflected, transmitted, or absorbed. Whether it is reflected, transmitted, or absorbed is decided by Russian roulette [3, 4] based on the bidirectional scattering distribution function (BSDF) of the surface.

A photon is traced until it is absorbed or hits the fictitious target surface located in front of the LGP, where the target surface is associated with a regular grid. The photons passing through each bin of the grid are counted and this counter is the estimator of the equilibrium photon flux density of the position on the LGP associated with the bin. Russian roulette technique can also be applied for the termination of tracing individual photons. The simulation is designed to loop through successive emisisons from the source surface until a prescribed accuracy level is attained or a maximum number of photons are emitted.

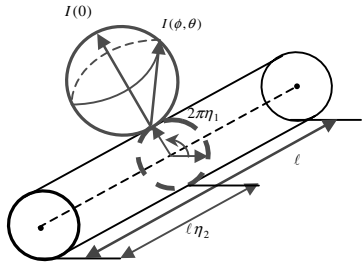


Fig. 2. Random emission of photons on the surface of CCFL

3.1 Photon Emission

Photons are emitted at random from the surface of CCFL in this simulation model. Using random numbers, a surface location for photon emission can be selected, and then another random number can be used to assign a direction of departure for the photon. Suppose that CCFL is a cylinder with radius r and length ℓ as shown in figure 2. Then it can be represented as a biparametric surface with each point represented by (u, v) where u is related to circumferential angle and v is related to the length. As the surface is symmetric along the circumferentail direction and uniform along the length, uniform randum numbers $\eta_1, \eta_2 \in [0,1]$ will give uniform emition point $(2\pi\eta_1, \ell\eta_2)$ of photons on the surface of CCFL.

It is also assumed that CCFL emits perfectly diffuse light energy. Therefore photon emission is equally probable in all directions. The direction of photon can be represented by a spherical coordinates (ϕ, θ) where ϕ is the circumferential angle and θ is the cone angle around the normal direction at emission point on the CCFL surface as in figure 2. Then, these angles are sampled as:

$$(\phi, \theta) = (2\pi\xi_\phi, \sin^{-1}\sqrt{\xi_\theta}) \tag{1}$$

where $\xi_\phi, \xi_\theta \in [0,1]$ are uniform random numbers.

3.2 Optical Properties of BLU Components

When a light ray is incident upon the interface between two transparent media, the incident ray is divided between a reflected ray and a refracted (transmitted) ray. The law of reflection and Snell's law predict the path of the resulting rays, and Fresnel's law of reflection predicts the amount of powers carried by each ray [8, 9]. In this paper, we assume that PMMA, the material of LGP, has wave-length independent radiative properties. But radiative properties of the PMMA are sensitive to direction, i.e., angle of incidence. For the angle θ_i of incidence, the refracted angle θ_t when a photon transmits to PMMA from air is calculated from the Snell's law:

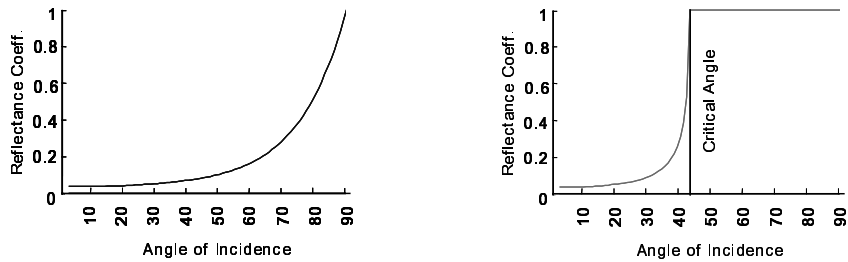
$$\eta_a \sin\theta_i = \eta_p \sin\theta_t \tag{2}$$

where $\eta_a \approx 1.0$ is the refractive index of air, and $\eta_p \approx 1.49$ is the index of PMMA.

The calculations of the fractions of the incident light that are reflected or transmitted for the interface depend on polarization of the incident ray. For the light polarized perpendicular or parallel to the interface, the reflection coefficients R_s, R_p are given by:

$$R_s = \left\{ \frac{\sin(\theta_i - \theta_t)}{\sin(\theta_i + \theta_t)} \right\}^2, \quad R_p = \left\{ \frac{\tan(\theta_i - \theta_t)}{\tan(\theta_i + \theta_t)} \right\}^2 \tag{3}$$

respectively. For the transmitted ray, the coefficient in each case is given by $T_s = 1 - R_s$ and $T_p = 1 - R_p$. For unpolarized light, the reflection coefficient becomes $R = (R_s + R_p)/2$.



(a) (b)

Fig. 3. Angular dependence of reflection coefficients: (a) air-to-PMMA transition (b) PMMA-to-air transition

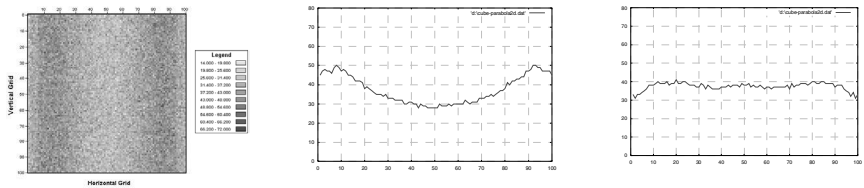
Figure 3 shows the reflectance coefficients for the air-to-PMMA and PMMA-to-air transitions, which are computed with Fresnel's equation, and they are comparable with the available experimental data. When a light moves from PMMA to air, all light is reflected if the incidence angle is above the critical angle that is 42°. This phenomenon is known as total internal reflection.

When a photon hits the surface of PMMA, Russian roulette is used to determine whether the photon is absorbed, reflected or transmitted [3, 4]. The surfaces of a lamp reflector and a reflection sheet are modeled with Phong model [9]. When a photon hits these surfaces, Russian roulette decides whether the photon is to be absorbed, reflected, or diffused. When the photon is reflected, the direction is importance sampled according to the BRDF of the surfaces. The surface of scattering ink dots is modeled with Lambert model. When a photon hit the surface, Russian roulette decides whether the photon is to be absorbed or diffused. For diffuse reflection, the direction of photon is sampled uniformly over the hemisphere as in equation (1).

3.3 Simulation

The fictitious target surface positioned on the front surface of LGP records the flux and the direction of the incident photons. For the following specific example, BLU for a notebook computer with 14″ LCD monitor is modeled to test the simulator. In this

type of BLU, two lamp holders are mounted at the left and right edges of the LGP and two lamps are enclosed in each lamp holder. A grid of size 100×100 is associated with the target surface and the number of photons passing through each bin of the grid is counted.



(a) (b) (c)

Fig. 4. The photon distribution detected on the target surface

Figure 4 (a), (b) shows a result of sample simulation, where 400,000 photons have been emitted from the source and all ink dots printed on the back surface of LGP have the same size. The count of photons in each cell of the grid is depicted in the figure 4 (a), where the lamps are positioned in the left and right sides of the grid. The result shown in figure 4 (b) corresponds to the cross section along the center line of (a). It shows high luminance on the area near to the edges and decreasing luminance getting closer to the center. Figure 4 (c) shows a cross section of a uniform illumination after optimization explained in the next section is applied to control the fill factor gradation of the ink dots.

4 Optimization

The purpose of ray tracing model described in previous section is to provide detailed prediction an optical performance of BLU. The performance is characterized quantitatively as a function of parameters which are defined to model the structure and optical properties of BLU components. The parameters many include the dimension of BLU components, the number and location of lamps, the shape of lamp reflector, the density (or fill factor gradation) and pattern of scattering ink dots, BSDF of scattering ink, refraction index of LGP, etc. Some parameters are assigned by fixed values during the design step and values for other parameters may change to achieve improved performance. Given an initial design of BLU by assigning some specific values to parameters modeling the structure of BLU, the ray tracing model can determine the performance of BLU. The next step is to analyze and enhance the design by adjusting some parameters to achieve improved performance. Numerical optimization techniques can be applied to produce an optimal design by automatically calculating the best values for the parameters.

Mathematically, the (unconstrained) optimization problem is to maximize (or, minimize) the objective function $f(\mathbf{x})$ where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ denotes a real n-vector of variables. In the ray tracing model of BLU, the variables are parameters to define the structure of BLU. The objective function is the performance of BLU, for example, the brightness and the luminance uniformity, etc. Among the parameters modeling

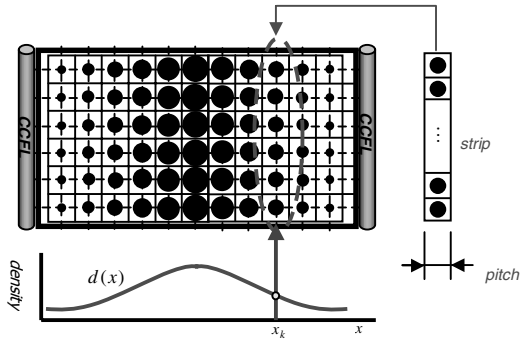


Fig. 5. The radius of circular ink dots can be computed from the density, fill factor gradation, function $d(x)$. The density function approximates the area of ink dots in a strip relative to the area of the strip, where x is the distance from the light source to the strip. The pitch, the width of strip, is less than 1mm

BLU, ink pattern is considerded to be the most important factor affecting the optical performance of BLU. Optimization with respect to other parameters seems likely to be amenable to the same approach described in this paper. In this paper, we consider the problem of optimizing the ink pattern to have a uniform distribution of luminaries on the front face of BLU. The unifom distribution is measured by the root mean square (RMS) values which are the counts of photons passing through each bin of the grid associated with the target surface.

The ink dots are arranged in a uniform grid of square or hexagonal cells and ink spots are positioned at centers of cells, as is shown in figure 5. The cell pitch, or the distance between the centers of two adjacent cells, depend on the printing technology and a typical pitch value for current technology is less than 1 milimeter. The uniformity of the light distribution on the BLU can be obtained by varying the size of the dot per unit area of the squared grid, or the fill factor gradation of the ink spots. In general, the shape of ink dots are circles and the size of ink dots in cells of a column of sqaure cells, called a strip, in the grid is the same. Therefore the fill factor gradation can be represented by a density function $d(x)$ which approximates the area of ink dots in a strip relative to the area of the strip, where x is the distance from the light source to the strip in the direction of light propagation. If a density funciton $d(x)$ is given, the diameter of ink dots in cells of a strip can be computed from the density function quickly.

Luminous intensity variations, as measured across the front surface of a LGP, determine the evenness of illumination. The ideal would be to have a flat illumination pattern so that the luminous intensity is the same at all points across the surface. However, in a good design there are typically minor luminous intensity variations across the surface. Careful selection of a density function design may produce an acceptable illumination pattern where the variations should not exceed 20% of the luminous intensity at center.

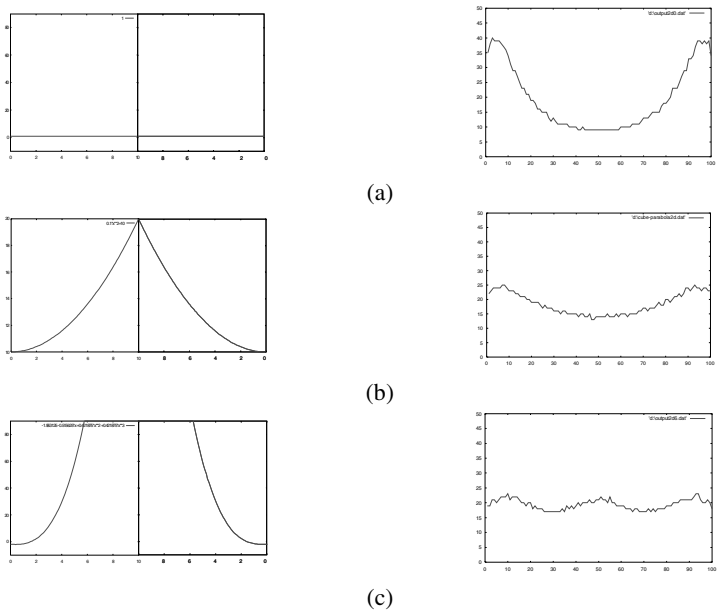


Fig. 6. Various density functions, in the left side, and the luminous intensity patterns, in the right side, that are produced by the ink patterns computed from the corresponding density functions

Figure 6 shows various density functions, in the left side, and the luminous intensity patterns, in the right side, that are produced by the ink patterns computed from the corresponding density functions. For example, figure (a) shows a luminious intensity pattern produced by a constant density function which generates very small and the same size ink dots across the grid cells. The pattern shows very low intensity in the center relative to the edges and should be avoided. Figure 6 (b), (c) show luminious intensity patterns produced by density functions which generate relatively large size ink dots in the center of the grid cells.

In this paper, we selected a polynomial of degree 3 to represent the density function. Then the objective function f represents the illumination uniformity across the front face of LGP, where the parameters are four coefficients of the density function controling the size of ink dots in the cell grid.

For the next step, optimization is needed to automatically compute the best values for the four parameters to achieve an optimal uniform illumination. We adapted the Nelder-Mead direct search method [7] to the optimal ink pattern design problem. At each iteration of the Nelder-Mead method, a simplex in n-dimensional space ($n = 4$ in our case) is constructed, and the function f in every vertex is evaluated. Based on the order of the observed function values in the vertices of the simplex, different operations can be taken to find better vertices. There are four operations on the simplex: reflection, expansion, contraction and shrinking. At each iteration, reflection, expansion, and contraction operations replace the worst vertex with the new better vertex, giving a new simplex. Shrinking operation selects the best vertex and

generates new other vertices closer to the best point than previous vertices. This step is repleated until some desired bound is obtained.

Standard coefficients for operations on the simplex to control the positions of the new vertices are used in the implementation even though a delicate choice of problem-dependent coefficients by some substantial level of work may improve efficiency of the implementation.

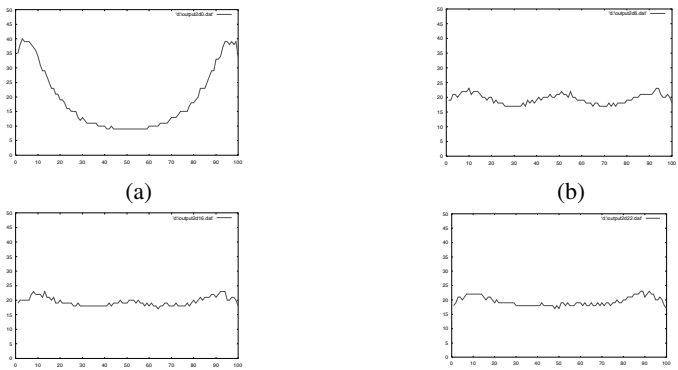


Fig. 7. Change of luminious intensity pattern generated by density function that corresponds to a simplex constructed after some iterations of the Nelder-Mead method. (a) initial density function which is a constant function (b) after 6^{th} iteration (c) after 16^{th} iteration (d) the final, after 22^{nd} iteration

Figure 7 shows the change of luminious intensity pattern generated by density function as the corresponding simplex rolles down to the optimal position in each iteration of the Nelder-Mead method. In this example, a constant density function is selected to assign the initial value of the four parameters of the objective function f.

One of the issues in the direct search method is the specification of convergence criteria. The direct search method performed well and remained popular to solve optimization problems for 30 years after it is devised in the early 1960s without a formal convergence proof. Torczon [8] discusses in detail the impossibility of developing general-purpose convergence tests for direct serach methods. In the design of BLU problem, we defined several termination conditions, for example, by assigning an appropriate value to the bound of the value f, on the size of simplex, etc. In a large number of experiments on different types of BLU, the direct search method consistently converges to an acceptable minimal solution within a reasonable number of iterations.

5 Conclusion

The design of BLU requires uniform illumination across the surface of BLU and high luminance enough to produce good contrast in a day environment. In this paper, an analysis method to predict the luminous intensity on the front face of BLU and an optimization method to get uniformity of illumination have been developed. The

analysis method of BLU is an application of Monte Carlo photon tracing. In conventional BLU, the uniformity of illumination is controlled by the pattern of diffusing ink spots. An optimal pattern of ink dots is searched by a method based on the simplex method by Nelder and Mead.

This analysis method by Monte Carlo simulation can be applied readily to evaluate the optical performace of other strutural type of BLU. An ink pattern printed on the back surface of LGP can be replaced with micro-prism shapes grooved in the back surface of LGP. The side lighting can be replace with bottom lighting that is welcomed for large-size BLU. In the bottom lighting, the number of lamps, distances between lamps, the geometric shape of lamp reflector are very crucial to generate the uniformity of illumination and the optimal values for these parameters can be determined by the optimization technique discussed in this paper.

References

1. Gebauer, M., Benoit, P., Knoll, P., Neiger, M.: Ray Tracing Tool for Developing LCD-Backlights, SID Digest 00 (2000) 558–561
2. Koyama, K.: Ray-Tracing Simulation in LCD Development, Technical Journal, Sharp Corp. (2002)
3. Pattanaik, S.N., Mudur, S.P.: Computation of Global Illumination by Monte Carlo Simulation of the Particle Model of Light. Proceedings of the 3rd Eurographics Workshop on Rendering (1992) 71–83
4. Jensen, H.W.: Realistic Image Syntehsis Using Photon Mapping, A K Peters (2001)
5. Teijido, J.M.: Conception and Design of Illumination Light Pipes, Ph.D. Thesis, University of Geneva, Switzerland (2000)
6. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vettering, W.T.: Numerical Recipes in C, Cambridge University Press, Cambridge (1988)
7. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization, Computer Journal, Vol. 7 (1965) 308–313
8. Blanis, C.A.: Advanced Engineering Electromagnetics, Jone Wiley & Sons (1989)
9. Glassner, A.S.: Principles of Digital Image Synthesis, Morgan Kaufmann Publishers (1995)
10. Torczon, V.: On the Convergence of Pattern Search Algorithm, SIAM J. Optimization, Vol. 7 (1997) 1–25

An Efficient Parameter Estimation Technique for a Solute Transport Equation in Porous Media

Jaemin Ahn[1], Chung-Ki Cho[2⋆], Sungkwon Kang[3], and YongHoon Kwon[1⋆⋆]

[1] Department of Mathematics, POSTECH, Pohang 790-784, South Korea
{gom,ykwon}@postech.ac.kr
[2] Department of Mathematics, Soonchunhyang University, Asan 336-745, South Korea, ckcho@sch.ac.kr
[3] Department of Mathematics, Chosun University, Gwangju 501-759, South Korea
sgkang@chosun.ac.kr

Abstract. Many parameter estimation problems arising in the solute transport equations in porous media involve numerous time integrations. An efficient adaptive numerical method is introduced in this paper. The method reduces the computational costs significantly compared with those of the conventional time-marching schemes due to the single time-integration, the spatial adaptiveness, and the $O(log(N))$ effects of the method, where N is the spatial approximation dimension. The efficiency and accuracy of the proposed algorithm is shown through a simple one-dimensional model. However, the methodology can be applied for more general multi-dimensional models.

1 Introduction

The movement of a contaminant through porous media such as a saturated aquifer is usually modelled by the transport equations together with an appropriate set of initial and boundary conditions in which various geophysical parameters such as hydraulic conductivities and dispersion coefficients are involved. The estimation of those parameters is one of the main concerns in hydrology[8,14]. To estimate the parameters, an appropriate optimization technique is needed. During the optimization process, numerous time-integrations need to be performed. These numerous integrations are the main obstacles in the estimation process. Therefore, we need an efficient algorithm for handling those obstacles.

In this paper, we consider a parameter estimation problem for the following one-dimensional transport equation

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - v(x)\frac{\partial c}{\partial x} + f(x,t), \quad (x,t) \in (0,\mathrm{X}) \times \mathbb{R}^+, \tag{1}$$

with the initial and boundary conditions

$$c(x,0) = 0, \quad c(0,t) = C_{\mathrm{L}}(t), \quad c(\mathrm{X},t) = C_{\mathrm{R}}(t). \tag{2}$$

⋆ The work of this author was supported by Soonchunhyang University, 2002.
⋆⋆ This paper is partially supported by Com^{2}MaC-KOSEF.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 847–856, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Here, $c(x,t)$ is the solute concentration at position x and time t, f is a source/sink term, C_{L} and C_{R} are given functions of time. The parameters D and v represent the dispersion coefficient and the transport mean velocity, respectively. The model (1)-(2) describes the movement of a solute through the groundwater flow. For simplicity, we assume that v is known and try to estimate D only. Let $\mathrm{Q} = \mathbb{R} = \{D\}$ be the parameter set. Under suitable conditions on the functions v, f, C_{L} and C_{R}, the model (1)-(2) has a unique solution in the class $\mathrm{C}^1(]0,\infty[;\mathrm{H}^1(0,\mathrm{X}))$[10], where H^1 denotes the usual Sobolev space. The solution of (1)-(2) with the parameter D will be denoted by $c(x,t;D)$. Then, the parameter estimation problem is to determine the parameter D from a set of observations of (a part of) the solution. Let T_o be a fixed time, $\{x^\alpha\}_{1\le\alpha\le n}$ a fixed set of points in $(0,\mathrm{X})$, and Δx a sufficiently small fixed positive number. Suppose we are given a set of n measurements $\{\omega^\alpha\}_{1\le\alpha\le n}$, where ω^α denotes the observation of the averaged solute concentration in the interval $[x^\alpha - \frac{1}{2}\Delta x, x^\alpha + \frac{1}{2}\Delta x]$. Then, our problem is

Problem P. Let $\tilde{\mathrm{Q}}$ be an admissible parameter subset of the parameter space Q. Given a set of measurements $\{\omega^\alpha\}$, find $D^* \in \tilde{\mathrm{Q}}$ which minimizes the cost functional $\mathrm{J} : \tilde{\mathrm{Q}} \to \mathbb{R}$ defined by

$$\mathrm{J}(D) = \sum_{\alpha=1}^{n} \left\{ \frac{1}{\Delta x} \left[\int_{x^\alpha - \frac{1}{2}\Delta x}^{x^\alpha + \frac{1}{2}\Delta x} c(x, \mathrm{T}_o; D)\, dx \right] - \omega^\alpha \right\}^2 .$$

2 Theory and Algorithm

The general theory of parameter estimation can be found in, for example, [2]. In this section, we describe the parameter estimation process briefly and derive our approximation algorithm.

We begin with considering an admissible parameter subset $\tilde{\mathrm{Q}}$ of Q. It is natural to assume that the dispersion coefficient is positive and bounded by a large constant. Thus, let $\tilde{\mathrm{Q}}$ be a compact interval in $\mathbb{R}^+$. It can be shown that the map from $\tilde{\mathrm{Q}}$ to $\mathrm{L}^2(0,\mathrm{X})$ defined by $D \mapsto c(\cdot, \mathrm{T}_o; D)$ is continuous. This implies the continuity of the cost functional J in Problem P, and, hence, we know, from the compactness of $\tilde{\mathrm{Q}}$, that Problem P has a solution. Now, suppose that we are given a numerical approximation scheme for (1)-(2), and, for each $N \in \mathbb{N}$, let $c^N(x,t;D)$ denote the corresponding finite dimensional approximate solution. Then, we obtain a sequence of finite dimensional problems approximating Problem P.

Problem $\mathbf{P^N}$. Let $\tilde{\mathrm{Q}}$ be an admissible parameter subset of the parameter space Q. Given a set of measurements $\{\omega^\alpha\}$, find $D^* \in \tilde{\mathrm{Q}}$ which minimizes the cost functional $\mathrm{J}^N : \tilde{\mathrm{Q}} \to \mathbb{R}$ defined by

$$\mathrm{J}^N(D) = \sum_{\alpha=1}^{n} \left\{ \frac{1}{\Delta x} \left[\int_{x^\alpha - \frac{1}{2}\Delta x}^{x^\alpha + \frac{1}{2}\Delta x} c^N(x, \mathrm{T}_o; D)\, dx \right] - \omega^\alpha \right\}^2 .$$

We hope that each problem P^N has a solution D^N and that the sequence $\{D^N\}$ converges to a solution of the original problem P. In fact, for a suitable choice of approximation scheme such as the Crank-Nicolson-Galerkin finite element scheme[9], we can prove the following.

[H1] For each $N \in \mathbb{N}$, there exists a solution $D^N \in \tilde{\mathrm{Q}}$ of Problem P^N.
[H2] There exists an increasing sequence $\{N_k\}$ in $\mathbb{N}$ such that the resulting subsequence $\{D^{N_k}\}$ converges to an element in $\tilde{\mathrm{Q}}$.
[H3] Suppose that $\{N_k\}$ is an increasing sequence in $\mathbb{N}$. If the corresponding subsequence $\{D^{N_k}\}$ converges to $D^* \in \tilde{\mathrm{Q}}$, then D^* is a solution to the original problem P.

A parameter estimation scheme satisfying the above conditions [H1]-[H3] is called the parameter estimation convergent one[2]. The essential estimates are the convergence of numerical solutions $\{c^N\}$ and the continuity of the maps $\tilde{\mathrm{Q}} \to \mathrm{L}^2(0, \mathrm{X})$, $D \mapsto c^N(\cdot, \mathrm{T}_o; D)$. For more details, see, for example, [2, 3] and references therein. There are many numerical optimization techniques for solving the finite dimensional minimization problems such as Problem P^N. Among them, we choose the Newton-Raphson(N-R) method[12] for simplicity. The method is following.

Starting with a suitable initial guess D_0 we generate the sequence $\{D_k\}$ by

$$D_k = D_{k-1} - \left(\frac{d\mathrm{J}^N}{dD}(D_{k-1}) \right)^{-1} \mathrm{J}^N(D_{k-1}), \quad k \geq 1. \tag{3}$$

Note that the derivative of the cost function J^N can not be computed exactly since it is defined via the discretized approximation c^N which is not explicitly known. Thus, we approximate the first derivative of J^N as

$$\begin{aligned} \frac{d\mathrm{J}^N}{dD}(D_{k-1}) \quad &\approx -2 \sum_{\alpha=1}^{n} \left\{ \frac{1}{\Delta x} \left[\int_{x^\alpha - \frac{1}{2}\Delta x}^{x^\alpha + \frac{1}{2}\Delta x} c^N(x, \mathrm{T}_o; D_{k-1})\, dx \right] - \omega^\alpha \right\} \cdot \\ &\frac{1}{\varepsilon} \frac{1}{\Delta x} \left[\int_{x^\alpha - \frac{1}{2}\Delta x}^{x^\alpha + \frac{1}{2}\Delta x} \left[c^N(x, \mathrm{T}_o; D_{k-1} + \varepsilon) - c^N(x, \mathrm{T}_o; D_{k-1}) \right] dx \right], \end{aligned} \tag{4}$$

where ε is a small positive real number. The algorithm for our parameter estimation process is stated as below. Let Tol be a tolerance for stopping the iteration and N_0 be the maximum number of iterations. The approximation of the first derivative of J^N in (4) is denoted by $d\mathrm{J}^N$.

Algorithm 2.1 (PE: Parameter Estimation)

Step 1. Set an initial guess $D_{cur} = D_0$.
Step 2. Solve the forward problem (1)-(2) with the parameter D_{cur}.
Step 3. Compute $\mathrm{J}^N(D_{cur})$.
Step 4. Set $\delta = |D_{cur}|$ and $count = 0$.
Step 5. While ($\delta >$ Tol) and ($count > N_0$), do Steps 6-11.

Step 6. Solve the forward problem (1)-(2) with the parameter $D_{cur}+\varepsilon$.
Step 7. Compute $d\mathrm{J}^N(D_{cur})$.
Step 8. Set $D_{next} = D_{cur} - (d\mathrm{J}^N(D_{cur}))^{-1}\mathrm{J}^N(D_{cur})$.
Step 9. Solve the forward problem (1)-(2) with the parameter D_{next}.
Step 10. Set $\delta = |D_{cur} - D_{next}|$ and $count = count + 1$.
Step 11. Set $D_{cur} = D_{next}$.

To obtain the approximation of the first derivative of J^N, for each iteration, we should solve the forward problem (1)-(2) twice; the one is for $c^N(x^\alpha, \mathrm{T}_o; D_{k-1}+\varepsilon)$ and the other is for $c^N(x^\alpha, \mathrm{T}_o; D_{k-1})$. Therefore, to terminate the iteration processes for the N-R method, we should solve the forward problem fairly many times. These solving processes are the main obstacles in the parameter estimation procedure. Hence, if we solve the forward problem efficiently, the total computational costs of the estimation process can be reduced significantly. It has been reported that the appropriate spatial discretization methods combined with the Laplace transform have advantages such as reducing the oscillation in the numerical solution and improving the computational efficiency due to single time-integration compared with the time-marching ones[1, 5,6,13]. In the following, we derive an efficient adaptive approximation scheme. It involves single time-integration, the spatial adaptiveness, and the $O(log(N))$ effects in the approximation dimension of the Laplace transforms.

Let $\{\Phi_i(x)\}_{1\le i\le N}$ be a basis for a spatial approximation space and try to find an approximate solution of the form $\sum_{j=1}^{N} c_j(t)\Phi_j(x)$. Take Laplace transformation to the model (1)-(2) to get

$$D\frac{\partial^2 \bar{c}}{\partial x^2} - v(x)\frac{\partial \bar{c}}{\partial x} + \bar{f}(x,s) = s\bar{c}, \quad (x,s) \in (0,\mathrm{X}) \times \mathbb{R}^+, \tag{5}$$

$$\bar{c}(0,s) = \bar{C}_{\mathrm{L}}(s), \quad \bar{c}(\mathrm{X},s) = \bar{C}_{\mathrm{R}}(s), \tag{6}$$

where s is the Laplace transform variable, and the bar($^-$) denotes the Laplace transformed function. Using the Galerkin formulation, equations (5)-(6) can be transformed into the following matrix-vector equation:

$$(A + sB)\bar{\mathbf{c}}(s) = \bar{\mathbf{b}}(s), \tag{7}$$

where A is the "stiffness" or "conductivity" matrix, B is the "mass" or "capacity" matrix, $\bar{\mathbf{c}}(s)=[\bar{c}_1(s), ..., \bar{c}_N(s)]^{\mathrm{T}}$ is the transformed vector of nodal concentrations, and $\bar{\mathbf{b}}(s)$ is the transformed vector containing the effects of source/sink terms and boundary conditions. The coefficients $\{c_j(t)\}_{1\le j\le N}$ is approximated from $\{\bar{c}_j\}$ by the inverse Laplace transformation as

$$c_j(t) \approx \frac{1}{T}\exp(\gamma t)\left[\frac{1}{2}\bar{c}_j(\gamma) + \sum_{k=1}^{\infty}\mathrm{Re}\left\{\bar{c}_j\left(\gamma + \frac{k\pi i}{T}\right)\exp\left(\frac{k\pi t i}{T}\right)\right\}\right], \tag{8}$$

where γ and T are suitably chosen constants (see Remark 1). In general, the infinite series (8) converges converge very slowly[4]. To accelerate the convergence

of the series, we apply the quotient-difference(q-d) scheme[11] to approximate the series. For notational simplicity, we write

$$p_j(z) = \sum_{k=0}^{\infty} a_k^{(j)} z^k,$$

where $a_0^{(j)} = (1/2)\bar{c}_j(\gamma)$, $a_k^{(j)} = \bar{c}_j(\gamma + ik\pi/T)$, and $z = \exp(i\pi t/T)$. Then, (8) reads $c_j(t) \approx (1/T)\exp(\gamma t)\mathrm{Re}\{p_j(z)\}$. Let

$$r_j(z) = d_0^{(j)}/(1 + d_1^{(j)} z/(1 + d_2^{(j)} z/(1 + \cdots)))$$

be the continued fraction corresponding to the series $p_j(z)$ and

$$r_j(z, L_j) = d_0^{(j)}/(1 + d_1^{(j)} z/(1 + \cdots + d_{L_j}^{(j)} z)) \tag{9}$$

be the L_j-th partial fraction of $r_j(z)$. Here, the coefficients $d_k^{(j)}$ of $r_j(z, L_j)$ is determined from the coefficients $a_k^{(j)}$'s of $p_j(z)$ for $k = 0, 1, \cdots, L_j$. Then, the time-dependent nodal concentration can be approximated as

$$c_j(t) \approx \frac{1}{T}\exp(\gamma t)\mathrm{Re}\{r_j(z, L_j)\}.$$

In (9), we do not know in advance how large L_j guarantees the sufficiently small error between $r_j(z, L_j)$ and its corresponding series. For each $c_j(t)$, the following stopping criterion can be considered:

$$|r_j(z, L_j^*) - r_j(z, L_j^* - 1)| < \mathrm{TOL}. \tag{10}$$

We call L_j^* the "computational optimal order". For the convection-dominated transport problems, we can expect that the computational optimal order L_j^* is small in a smooth region, and is large in a steep gradient region. This automatic determination of L_j^*'s has the effects of the spatial adaptiveness and reduces the computational costs. It has been reported[1] that $\left[\max_{1\le j\le N} L_j^*\right] \sim O(\log N)$. Thus, the float-number of operations of this method is $O((\log N)^2 N)$, while the conventional time-marching schemes require $O(N^2)$ operations. We now ready to state our adaptive algorithm.

Algorithm 2.2 (ALTG: Adaptive Laplace Transform Galerkin Method)

Step 1. For $j = 1, ..., N$, set $fl_j = False$.
Step 2. Set $flag = N$.
Step 3. Solve the linear system (7) for $s = s_0, s_1$, and s_2.
Step 4. For $j = 1, ..., N$, do Steps 5-6.
Step 5. Determine $d_0^{(j)}, d_1^{(j)}$, and $d_2^{(j)}$ by the q-d algorithm.
Step 6. Evaluate $r_j(z, 1)$ and $r_j(z, 2)$.
Step 7. Set $L = 2$.
Step 8. While $flag \neq 0$, do Steps 9-17.
Step 9. Set $L = L + 1$.

Step 10. Solve the linear system (7) for $s = s_L$.
Step 11. For $j = 1, ..., N$, do Steps 12-17.
Step 12. If $fl_j = False$, do Steps 13-17.
Step 13. Determine $d_L^{(j)}$ by the q-d algorithm.
Step 14. Evaluate $r_j(z, L)$.
Step 15. If $|r_j(z, L) - r_j(z, L-1)| <$ TOL, do Steps 16-17.
Step 16. Set $fl_j = True$ and $flag = flag - 1$.
Step 17. Set $c_j(t) = (\exp(\gamma t)/T)\mathrm{Re}\{r_j(z, L)\}$.

Remark 1. For the choice of the parameter γ in equation (8), Crump[4] proposed $\gamma = \alpha - \log(E_R)/2T$, where α is a number slightly larger than $\max_j\{\mathrm{Re}\,(P) : P$ is a pole of $\bar{c}_j(s)\}$, $\bar{c}_j(s)$ is the Laplace transformed function of $c_j(t)$, and E_R is the relative discretization error tolerance. It is known that $\alpha = 0$, $E_R = 10^{-6}$, and $T = 0.8\mathrm{T_f}$, where $\mathrm{T_f}$ is the final simulation time, are adequate for general purpose[13], i.e.,

$$\gamma = -\log(E_R)/1.6\mathrm{T_f}. \tag{11}$$

3 Numerical Results

To show the parameter estimation convergence and the efficiency of our algorithm, we consider the following example[7].

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - v\frac{\partial c}{\partial x}, \quad (x,t) \in \mathbb{R}^+ \times \mathbb{R}^+, \tag{12}$$

$$c(x,0) = 0, \quad c(0,t) = C_\mathrm{L}, \quad \lim_{x\to\infty} c(x,t) = 0. \tag{13}$$

The analytic solution for (12) - (13) is given by

$$c(x,t) = C_\mathrm{L}\frac{1}{2}\left\{\mathrm{erfc}\left(\frac{x-vt}{2\sqrt{Dt}}\right) + \exp\left(\frac{vx}{D}\right)\mathrm{erfc}\left(\frac{x+vt}{2\sqrt{Dt}}\right)\right\}, \tag{14}$$

where $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x) = 1 - (2/\sqrt{\pi})\int_0^x e^{-u^2}du$. For our simulation, X, the length of the soil column, and $\mathrm{T_f}$, the final time or maximum simulation time, were chosen so that the solution (14) does not reach X under a sufficiently small tolerance limit, for example, 10^{-30} mg/L for the time interval $[0, \mathrm{T_f}]$. Thus, we may consider the following model:

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - v\frac{\partial c}{\partial x}, \quad (x,t) \in (0,\mathrm{X}) \times \mathbb{R}^+, \tag{15}$$

$$c(x,0) = 0, \quad c(0,t) = C_\mathrm{L}, \quad c(\mathrm{X},t) = 0. \tag{16}$$

For the numerical simulation, X = 50 m, $v = 0.1$m/day, $C_\mathrm{L} = 1.0$ mg/L, $\mathrm{T_f} = 250$ days, and the true dispersion coefficient $D = 0.005043$ m^2/day were chosen. Figure 1 shows the corresponding "exact" or "analytic" solution (14) at $t = \mathrm{T_f}$. The observation data $\{\omega^\alpha\}$ are collected by using the analytic solution (14) at

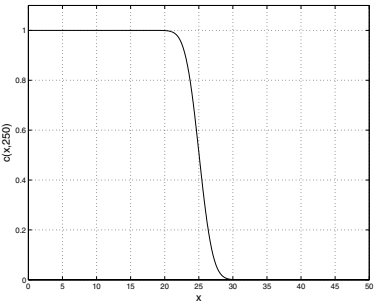


Fig. 1. Analytic solution at $t = 250$ days.

Table 1. OLS-Error

N	PE-ALTG	PE-FEMCN
128	1.75538e-03	1.88003e-03
256	4.72320e-04	3.91737e-04
512	8.85100e-05	8.97073e-05
1024	3.15797e-05	3.17310e-05
2048	2.89067e-05	2.88446e-05

Table 2. $|(D - D^N)/D|$

N	PE-ALTG	PE-FEMCN
128	2.16180e-02	2.19636e-02
256	3.24526e-03	3.74501e-03
512	1.17467e-03	1.15846e-03
1024	2.44418e-04	2.39746e-04
2048	8.70740e-05	8.98230e-05

the final time $\mathrm{T}_o = \mathrm{T_f}$ at uniformly distributed 40 observation points $\{x^\alpha\}$, where $x^\alpha = \alpha \mathrm{X}/41$, $\alpha = 1, \cdots, 40$. We assume that Δx is sufficiently small so that the averaged solute concentration in the interval $[x^\alpha - \frac{1}{2}\Delta x, x^\alpha + \frac{1}{2}\Delta x]$ can be regarded as the point value at x^α.

In the following we compare the numerical results of our method with those of a typical conventional method. Our method(PE-ALTG) uses the Galerkin method, Algorithm 2.1, and Algorithm 2.2, and the other(PE-FEMCN) uses the Galerkin method with the Crank-Nicolson time-stepping together with Algorithm 2.1. Given a spatial approximation dimension $N \in \mathbb{N}$, the time step Δt in PE-FEMCN was chosen such that the Courant number $Cr = v\Delta t/h = 0.1 \leq 1$, where $h = \mathrm{X}/N$. In applying PE-ALTG, the parameter $E_R = 10^{-6}$ was chosen in equation (11) and the tolerance TOL for the stopping criterion (10) was chosen as

$$\mathrm{TOL} = \frac{T}{e^{2\gamma T}} \frac{1}{N^2}$$

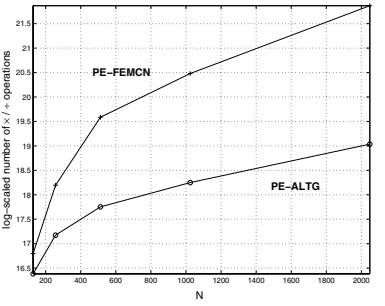


Fig. 2. Computational costs for PE-ALTG & PE-FEMCN

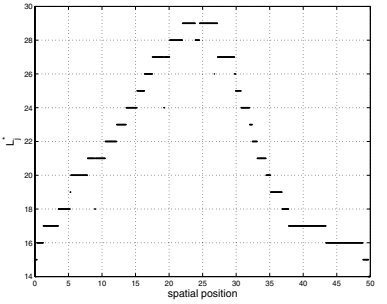


Fig. 3. Spatial distribution of L_j^* for $N = 1024$.

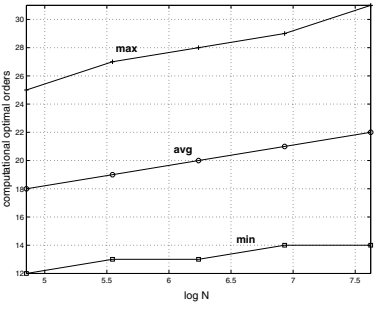


Fig. 4. Growth of computational optimal orders

so that the local error for PE-ALTG becomes

$$|c_j(\mathrm{T_f}) - \frac{e^{\gamma \mathrm{T_f}}}{T}\mathrm{Re}\left\{r_j(z, L_j^*)\right\}| \approx |\frac{e^{\gamma \mathrm{T_f}}}{T}\mathrm{Re}\left\{r_j(z, L_j^* - 1) - r_j(z, L_j^*)\right\}| \leq \frac{1}{N^2}.$$

For the optimization process (Algorithm 2.1), we started with the initial guess $D_0 = 0.01$. The tolerance Tol and the maximum number N_0 of iterations were chosen as 10^{-6} and 20, respectively. Table 1 and Table 2 show the OLS-Error and the relative error of the estimated parameters, respectively.

It is easy to see that the both methods have the parameter estimation convergence properties and the similar accuracy. Here, the OLS-Error means the output-least-squared(OLS) error

$$\left\{ \sum_{\alpha=1}^{40} \left[c^N(x^\alpha, \mathrm{T}_o; D^N) - \omega^\alpha \right]^2 \right\}^{\frac{1}{2}}.$$

Figure 2 shows the log-scaled total computational costs for the two parameter estimation schemes. By the computational cost we mean the number of multiplications and divisions as usual.

It is easily observed, from Figure 2, that the total computational costs for the PE-ALTG scheme have been reduced significantly(for example, approximately 89% for $N = 1024$ and 94% for $N = 2048$) compared with those for the PE-FEMCN scheme. It is due to the single time-integration, the spatial adaptiveness, and the $O(log(N))$ effects of our algorithm as mentioned before. Figure 3 shows a typical spatial distribution of the computational optimal orders. For this figure, the spatial approximation dimension was chosen as $N = 1024$. From Figure 1, we see that the steep gradient region appears approximately near $x = 25$m. In Figure 3, the computational optimal orders L_j^* appear to be high in the steep gradient region and low in the smooth regions. Figure 3 shows clearly the spatial adaptiveness of our algorithm. Figure 4 shows the maximum, average, and the minimum of the computational optimal orders. It is easy to see that they grow linearly with respect to $log(N)$.

4 Concluding Remarks

The transport equation is considered as a mathematical model for solute transport in porous media. We developed a fast approximation scheme(adaptive Laplace transform Galerkin technique) for estimating geophysical parameters. Numerical experiments show the efficiency and accuracy of our scheme. This technique can be applied to more general higher-dimensional problems.

References

1. Ahn, J., Kang, S., Kwon, Y.: A flexible inverse Laplace transform algorithm and its application. Computing **71** (2003) 115–131.
2. Banks, H. T., Kunisch, K.: Estimation Techniques for Distributed Parameter Systems. Birkhäuser, Boston (1989).
3. Cho, C.-K., Kang, S., Kwon, Y.: Parameter estimation for an infiltration problem. Comp. Math. Appl. **33** (1997) 53–67.
4. Crump, K. S.: Numerical inversion of Laplace transform using Fourier series approximation. J. Assoc. Comput. Mach. **23** (1976) 89–96.
5. Elzein, A.: A three-dimensional boundary element/Laplace transform solution of uncoupled transient thermo-elasticity in non-homogeneous rock media. Commun. Numer. Meth. Engng. **17** (2001) 639–646.

6. Farrell, D. A., Woodbury, A. D., Sudicky, E. A.: Numerical modelling of mass transport in hydrogeologic environments: performance comparison of the Laplace transform Galerkin and Arnoldi modal reduction schemes. Advances in Water Resources **21** (1998) 217–235.
7. Freeze, R. A., Cherry, J. A.: Groundwater. Prentice-Hall, N. J. (1979).
8. Giacobbo, F., Marseguerra, M., Zio, E.: Solving the inverse problem of parameter estimation by genetic algorithms: the case of a groundwater contaminant transport model. Annals of Nuclear Energy. **29** (2002) 967–981.
9. Hossain, M. A., Miah, A. S.: Crank-Nicolson-Galerkin model for transport in groundwater: Refined criteria for accuracy. Appl. Math. Compu. **105** (1999) 173–181.
10. Pazy, A.: Semigroups of Linear Operators and Applications to Partial Differential Equations, Springer-Verlag, New York (1983).
11. Rutishauser, H.: Der Quotienten-Differenzen-Algorithmus, Birkhäuser Verlag, Basel (1957).
12. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis, Springer-Verlag, New York (1993).
13. Sudicky, E. A.: The Laplace transform Galerkin technique: A time continuous finite element theory and application to mass transport in groundwater. Water Resour. Res. **25** (1989) 1833–1846.
14. Wai, O. W. H., O'Neil, S., Bedford, K. W.: Parameter estimation for suspended sediment transport processes under random waves. The Science of The Total Environment. **266** (2001) 49–59.

HierGen: A Computer Tool for the Generation of Activity-on-the-Node Hierarchical Project Networks

Miguel Gutiérrez[1], Alfonso Durán[1], David Alegre[1], and Francisco Sastrón[2]

[1] Departamento de Ingeniería Mecánica, Universidad Carlos III de Madrid
Av. de la Universidad 30, 28911 Leganés (Madrid), Spain
{miguel.gutierrez,alfonso.duran,david.alegre}@uc3m.es
http://www.uc3m.es/uc3m/dpto/dpcleg1.html
[2] DISAM, Universidad Politécnica de Madrid
c/ José Gutiérrez Abascal 2, 28006 Madrid, Spain
sastron@etsii.upm.es
http://www.disam.etsii.upm.es

Abstract. Hierarchical project networks are commonly found in many real life environments. Furthermore, currently there is a significant trend towards adopting a hierarchical approach in planning and scheduling techniques. However, no generator currently exists that automates the generation of hierarchical networks to facilitate the simulation of these environments and the testing of these techniques. The network generator presented in this paper is specifically aimed at fulfilling this need. HierGen generates, stores in a database and graphically represents Activity-on-the-Node (AoN) hierarchical networks with the desired number (or with a number drawn from the desired statistical distribution) of both aggregated and detailed activities, as well as the desired number of precedence relations linking them. It can also generate more conventional, non-hierarchical networks; in that case, it can perform a "sequential aggregation", grouping together those activities whose precedences come only from previous aggregated activities.

1 Introduction

Realistic simulation of project oriented environments requires the capability to model and generate series of project networks that are sufficiently similar to the projects generally found in these environments. Appropriate project network generators could provide this capability. The growing scientific field of project planning and scheduling techniques [1], [2], also relies on network generators for creating project networks on which these techniques can be tested (through simulation or otherwise). Several such generators have been proposed in the literature in the last decade, including [3], ProGen [4], ProGen/Max [5], DAGEN [6] and RanGen [7].

These network generators create one-level or "flat" project networks. However, in many real life environments, projects tend to have a hierarchical structure, with aggregated activities forming a first-level aggregated network, while each of these aggregated activities can be further decomposed (WBS) into a more detailed sub-network [8, p.150]. Analogously, given the current general trend towards

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 857–866, 2004.
© Springer-Verlag Berlin Heidelberg 2004

hierarchically oriented planning and scheduling techniques, testing and simulating these techniques would require the generation of sets of hierarchical project networks.

However, no generator currently exists that automates the generation of hierarchical networks to facilitate the testing of these techniques. The network generator whose algorithm is presented in this paper is aimed at fulfilling this need. HierGen is a computer tool that generates, stores in a database and graphically represents Activity-on-the-Node (AoN) hierarchical networks.

In the next section previous contributions in the area of network generators are briefly discussed. Then, the need for and the usefulness of the proposed HierGen tool is justified through the analysis of hierarchical networks and of hierarchical planning and scheduling techniques. Afterwards, the basic components and the algorithms of HierGen are described. The conclusions section summarizes the main contributions.

2 Network Generators

The literature on project network generators is relatively recent and scarce. Demeulemeester, Dodin and Herroelen published in 1993 [3] the description of a simplified generator for strongly random activity-on-the-arc project networks. One of the best known generators is ProGen, a generator of activity-on-the-node project networks described by Kolisch, Sprecher and Drexl in 1995 [4], that allows users to specify topology parameters; it was later extended to ProGen/Max by Schwindt [5].

Based on the definition of the Complexity Index (CI) by Bein, Kamburowski and Stallman [9], Agraval, Elamaghraby and Herroelen [6] developed the activity-on-the-arc project network generator DAGEN, in which the user can specify the Complexity Index. In 2003, Demeulemeester, Vanhoucke and Herroelen published the description of RanGen [7], a random network generator of AoN project networks that conform to preset values of complexity measures including the Order Strength (OS).

3 Hierarchical Networks and Hierarchical Planning and Scheduling

In many project oriented companies, hierarchical networks appear naturally. Prior to the execution of a customer order there is a quotation stage in which approximate budget and due date are determined. In this stage the order is considered as a roughly defined network of aggregated activities. If the order is finally accepted, then the engineering department, if necessary, adjusts the rough project network, and in any case refines each aggregated activity into more detailed activities, which in due time can constitute a network with precedence relations [8, p. 150]

In some environments the aggregated network is merely a sequence of aggregated activities. Van der Merwe describes how, in the detailed network of pure engineering projects, a lineal network can be recognized, encompassing four stages (aggregated activities): feasibility, design, construction and closing [10]. Each stage consists of a network of tasks. Van der Merwe states that for the sake of project control the hierarchical consideration of the project network has considerable advantages [10].

A valuable application for a generator for these hierarchical project networks would be supporting the simulation techniques widely used in production environments in decisions such as the acquisition of a new expensive resource, the determination of maintenance policies or the evaluation of the consistency of a plan against incidences (breakdowns, absenteeism, supply delays …). One application of special relevance would be the determination of the best due date algorithm [11].

However, in spite of the evidence of the existence of those hierarchical structures, the research concerning project planning and scheduling has focused almost exclusively on providing algorithms for the detailed network scheduling [1], [2]. Analogously to what happened with many well defined optimization problems, the need to test algorithms against a collection of problem instances motivated the development of the network generators described in the former section.

In the last few years there has been a trend towards a hierarchical approach to project planning [12], [13], [14]. This trend is being strongly reinforced by a general tendency in algorithmic research: "Integrating planning and scheduling is a hot research topic, especially in the planning community" [15]. Bartak highlights the different approaches historically taken to solve the planning and scheduling problems, despite of their similarities, and notices both the wide attention paid to the planning problems by the Artificial Intelligence (AI) community and the long tradition of the Operations Research (OR) community in studying scheduling problems [16].

With regard to the integration of planning and scheduling, while in the AI field the interest is quite recent [17], in the OR community, since the pioneer work of Hax and Meal [18] the parallelism between planning and scheduling when applied to production tasks has yielded a number of proposals dealing with the so-called Hierarchical Production Planning (HPP) [19], [20] (see [21] for a recent review). However, the practical application of these proposals has been of little significance, due mainly to the complexity and the problem-specific formulation of the models [21].

In the last few years, there has been a breakthrough in HPP, with the combined AI-OR approach. Particularly, the modelling capability of Constraint Programming [22] makes it easy to extend the production models to the project environments [23], so it can be expected that in the near future the project planning software will be enhanced with more sophisticated hierarchical planning algorithms. In summary, in addition to its interest for simulation purposes, there is a growing need for a generator of hierarchical project networks in order to test the increasing number of hierarchical project planning algorithms.

4 HierGen Basic Components

Similarly to previous generators, the network generator whose algorithm is presented in this paper, HierGen, is restricted to networks in which the relations are all precedence relations without delay. Nodes will be numbered in such a way that a precedence relation can only exist from node i to node j if $i < j$.

Similarly to Demeulemeester et al. [7], and utilizing the same notation for the sake of clarity, a project network in AoN format shall be represented by G=(N,A), where the set of nodes, N, represents activities and the set of arcs, A, represents precedence constraints by an upper triangular precedence relations matrix without the diagonal.

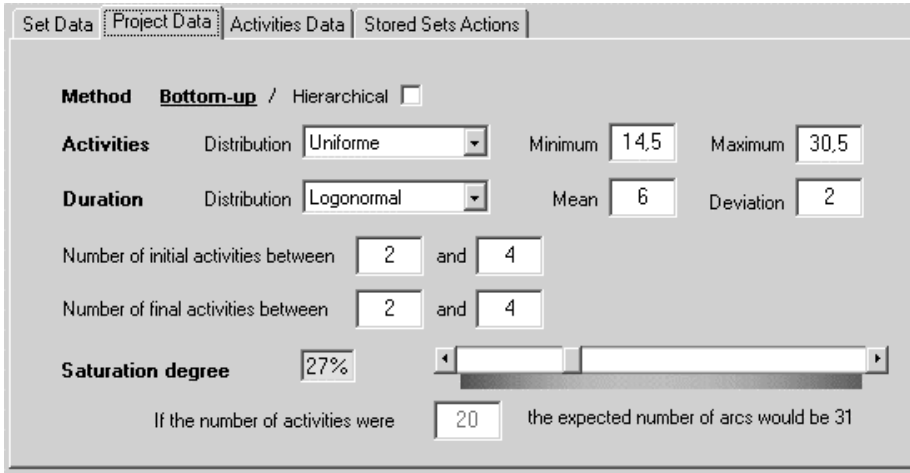


Fig. 1. HierGen screen used to define initialization parameters

This binary precedence matrix (PM) denotes whether or not a precedence relation exists between two nodes. If node *i* is a predecessor (either direct or transitive) of node *j*, then PM_{ij}=1; otherwise, it equals zero. It is worth highlighting that by reading this matrix row-wise it shows, in row i, all the direct and transitive successors of node i; by reading it column-wise, is shows, in column j, all predecessors of node j.

The Arc matrix contains the arcs themselves, i.e., if there is a precedence relation from node *i* to node *j*, then Arc_{ij}=1; otherwise, it equals zero. Arc can be considered a subset of PM containing only direct successors. Additionally, the Redundant matrix shows those arcs that would be redundant if added: if a precedence relation from node *i* to node *j* would be redundant if added, then $Redundant_{ij}$=1; otherwise, it equals zero.

User-definable parameters include: Nact: number of activities in the project; Narc: number of arcs in the AoN network; Ninitial: number of initial activities; Nfinal: number of final activities.

To allow the generation of sets of project networks, rather than stipulating a specific value for these parameters, users can choose among several statistical distributions and specify the distribution parameters (mean, range …) (see Fig. 1): specific parameter values are then stochastically drawn from those distributions.

To allow users to generate projects with the desired degree of complexity, rather than directly specifying Narc, users choose a "saturation degree" for the number or arcs, within the range of attainable values. As will be explained later, the tool assists the user, through a scroll bar control featured with a colour code (see Fig. 1), in choosing this value through a bar along which the user positions a handle; the bar encompasses the range of attainable values for this Narc over Nact ratio.

5 Bottom-Up Generation and Aggregation Algorithm

HierGen users can choose between a bottom-up "flat" one-level project network generation module, described in this section, and a top-down hierarchical network

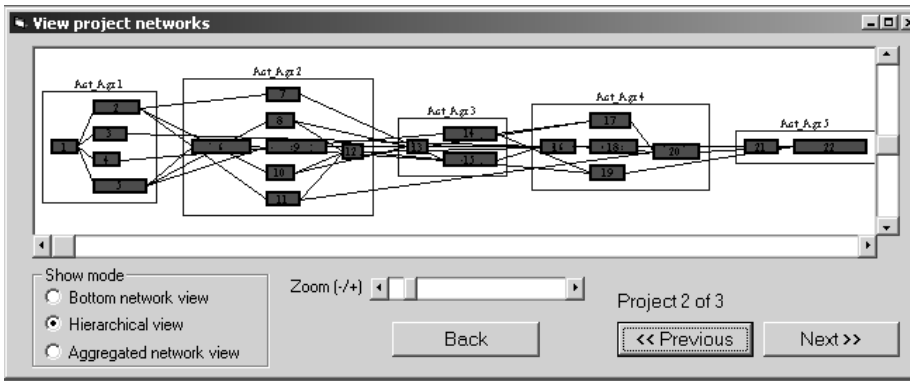


Fig. 2. Horizontally aggregated network

generation module. Activities in the "flat" network can then be horizontally aggregated.

The algorithm involves nine steps, inspired in [16], that are executed each time a project network instance is created, which are described below (see Fig.2 for an example as drawn by the tool):

1. Initialize parameters. The values of Nact, Ninitial and Nfinal for this project network instance are stochastically drawn from from the probability distributions selected by the user. Project networks are constructed so that each "initial" activity receives only one precedence link, which comes from the first activity, and each "final" activity has only one forward precedence link, to the last activity.
   Regarding Narc, as mentioned above, rather than asking the user to set a specific value for Narc, which would often not be attainable, the user chooses, with the help of HierGen, a ratio related to the arc saturation in the project. A value of 0% would imply an Narc equal to the minimal theoretically possible value, i.e. Narc=Nact-1, corresponding to a completely linear project (a most unlikely structure). A value of 100% would indicate the maximum theoretically possible value.
   However, given the stochastic way in which project networks are gradually built by the generator, it is very unlikely, for any given network instance, that it adopts the peculiar structure required for these extreme values. As an illustration, the chances that the project turns out to be linear, therefore allowing Narc to become its minimum value, Nact-1, is p=1/(Nact-1)!; for Nact=10 that means p=2,75·10-6. Therefore, a series of experiments have been conducted to determine, through statistical regression, an estimation equation for the minimum and the maximum number of arcs that are attainable, with a 50% probability, in a project network instance of a given number of activities Nact. The tool assigns the saturation value of 20% to this minimum number of arcs and 80% to the maximum number of arcs.
2. Initialize matrices. The precedence matrix (PM), the Arc matrix (Arc) and the Redundant matrix (Arc) are created as blank Nact x Nact matrices (in fact, upper triangular matrices).
3. Create arcs linking the first activity with the initial activities and those linking final activities with the last activity. A link is created showing a precedence relationship from the first activity in the project network to the Ninitial following activities.

This implies setting Arc_{1j}=1 and PM_{1j}=1 for j=(2, Ninitial+1). A link is created showing a precedence relationship from each of the Nfinal activities to the last activity. This implies setting Arc_{iNact}=1 and PM_{iNact}=1 for i=(Nact-1-Nfinal, Nact-1) .

4. Create a backwards link for each activity. Since all activities except the first one must have a backwards link, Arc_{ij}=1 is first created for each activity j, j= (Ninitial+2, Nact-Nfinal-1). Activity i, where this precedence relation to activity j starts, is randomly drawn from the range 2, i-1. Final activities must also have a backwards link, but it cannot come from another final activity; therefore, a backwards link Arc_{ij}=1 is then created for each activity j, j= (Nact-Nfinal, Nact-1), where i is randomly drawn from the range (2, Nact-Nfinal-1). While updating the precedence matrix (PM) each time a new link Arc_{ij}=1 is created, not only should PM_{ij} be set to 1, but also all direct and indirect predecessors of activity i (column i) should be added as predecessors of j (column j). That is, for k=(1,i-1), if PM_{ki}=1, set PM_{kj} =1. Up to this point in the algorithm there is no need to check for redundancy.
5. Create a forward link for each activity lacking successors. Since all activities except the last one must have a forward link, the algorithm now scans matrix Arc looking for empty rows. For each activity i without successors (blank i row), the algorithm then determines, for j=(i+1, Nact-1), if an eventual ij link (Arc_{ij}=1) would be redundant, i.e., if $Redundant_{ij}$=1. That requires updating the values in matrix Redundant through the corresponding HierGen module. Based on the information stored in matrices Arc and PM on the existing activities, precedence relations and indirect precedences, the algorithm in that module verifies, for each ij combination in the upper triangular Redundant matrix, if an eventual ij link would incur in any of the four redundancy scenarios listed below:
   - Activity j is a successor of activity i.
   - Any predecessor of activity i is a predecessor of activity j.
   - A direct successor of j is a successor of i.
   - For any k=(j+1, Nact), activity k is a successor of j, and a direct predecessor of activity k is also a predecessor of activity i.

   Then, among those activities j that fulfil j=(i+1, Nact-1) and $Redundant_{ij}$=0, one is randomly drawn, and an ij relation is created through updating matrices PM and Arc as described before. If no activity j fulfils both conditions, then an activity j is randomly chosen from j=(i+1, Nact-1), and a redundant relation ij is created.
6. Remove redundant links. Since the previous step might create redundant arcs, these should now be eliminated. The HierGen module that eliminates these redundant arcs scans all ij arcs (Arc_{ij}=1), tentatively eliminating it, then verifying, as described above, if it would be redundant if reinstated. If it is redundant, the algorithm verifies whether activity i has any other successor and activity j has any other predecessor, in which case relation ij can be eliminated.
7. Add new links until Narc is reached. At this point, the minimum number of arcs has been included. If this number is above the desired Narc, that target Narc can not be attained in this instance of the project network. If this number is smaller than Narc, then new ij relations are added until Narc is reached.

   The ij pairs are randomly drawn from among those that simultaneously fulfil: i=(2,Nact-Nfinal-2), j=(Ninitial+2, Nact-1), i<j, $Redundant_{ij}$=0. If no ij pair fulfils these conditions and the desired Narc has not yet been reached, that target Narc can not be attained in this instance of the project network.

8. Calculate activity durations. The duration for each activity is then stochastically drawn from the probability distribution specified by the user.
9. Horizontally aggregate activities. Once all activities and relations have been generated, activities can then be horizontally aggregated. One useful application of this aggregation is that it facilitates drawing the project network in a visually useful format. The algorithm aggregates together, in the first aggregated activity, the first activity and those activities whose only predecessor is the first activity (i.e., the initial activities). Successive aggregate activities encompass all those activities whose predecessors are contained in previous aggregate activities.

6 Hierarchical Generation Algorithm

In addition to the creation of "flat" projects whose activities can then be horizontally aggregated, HierGen offers the functionality of creating hierarchical project networks. Basically, the hierarchical generation algorithm operates in two phases. It first generates a network for the aggregated activities, and then generates a sub-network for the activities that make up each aggregated activity. Some differences with the bottom-up algorithm described above are:

- Additional, equivalent parameters and matrices are defined for the aggregated level, such as NactAgr and ArcAgr.
- No "Initial" or "Final" activities are contemplated. Since both the aggregated network and each of the sub-networks are now much smaller, imposing the restriction of a specific shape (just one predecessor in the initial activities, just one successor in the final activities) does not render project networks that are more similar to the real ones.
- Two alternative topologies are supported. The user must decide if, in the sub-network that corresponds to each aggregated activity, there will be one "opening" activity and one "closing" activity, so that any precedence relations among aggregated activities are implemented through precedence relations linking the closing activity within the first aggregated activity with the opening activity of the second aggregated activity. Alternatively, precedence relations among aggregated activities can be implemented through precedence relations among any of the activities of their corresponding sub-networks.

The algorithm involves 11 steps, most of which (3, 4, 5, 6, 7, 8, 11) follow, at the aggregated level, the same logic as in the bottom-up algorithm, and are therefore not explained. In the rest, differences are highlighted below. Fig. 3 shows an example, as drawn by the tool.

1. Initialize Parameters. NactAgr is stochastically drawn (it can be as low as 1). The number of activities within each aggregated activity is then stochastically drawn. The total number of activities is found by adding the number in each sub-network. The number of arcs is selected by the user in the same manner as in the bottom-up algorithm, by selecting a saturation ratio with the help of the HierGen assistance tool. Attainable maximum number of arcs is heavily influenced by the choice made on whether relations will be restricted to opening and closing activities.
2. Create a backwards link for each aggregated activity.
3. Create a backwards link for each aggregated activity.

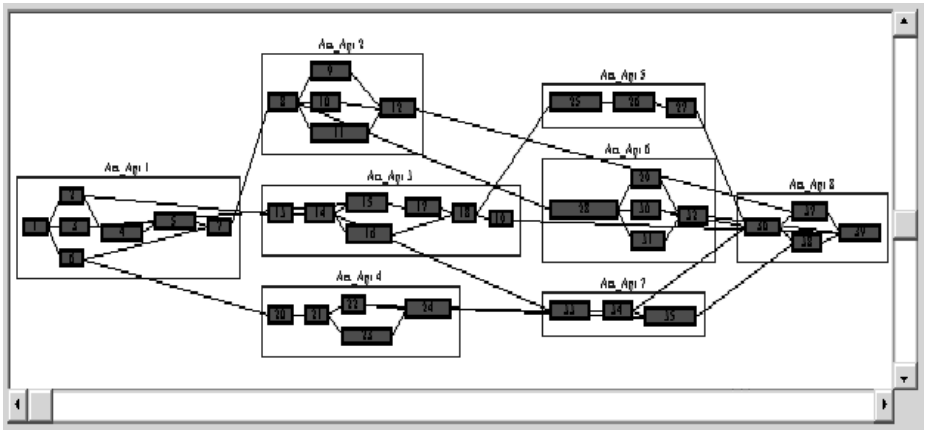


Fig. 3. Hierarchical project network

4. Create a forward link for each aggregated activity lacking successors.
5. Remove redundant links.
6. Create a backwards link for each activity within each aggregated activity.
7. Create a forward link for each activity lacking successors within each aggregated activity.
8. Remove redundant links.
9. Create links for the first and the last activity of each aggregated activity. If the topology based on opening and closing activities has been chosen, for each relation among aggregated activities a relation is created linking the closing activity of the first aggregated activity with the opening activity of the second aggregated activity. Otherwise, at least one backward link from the opening activity and one forward link from the closing activity of each aggregated activity are created, subject to the condition that they only link activities whose aggregated activities are linked.
10. Add new links until Narc is reached. Same logic as in the bottom-up algorithm, subject to the condition that the new relations only link activities whose aggregated activities are linked.
11. Calculate activity durations.

7 Completion of the Network Generation

To complete the database records of the projects and activities, some complementary but essential data are generated. The generation of resource requirements is described in more detail since it is a specific characteristic of hierarchical project networks.

Analogously to the activity hierarchy, there is also a hierarchy between resources. The experience of the authors with project oriented companies and the literature review have determined how resource requirements have been modelled in HierGen. First, the aggregated resource requirements are generated. It is very frequent that the criteria for grouping resources into an aggregated resource has a close relationship with the activity grouping, since typically an aggregated activity requires a small number of aggregated resources. Furthermore, it is very usual that the aggregated

activities require only one or two aggregated resources (those requiring quality inspection). The generator draws a number from a uniform distribution whose range is determined by the user, so it can be forced to be a unique or a small number, but also allows the possibility of not being so restrictive.

The requirements of the detailed activities are then generated, coherently with the former assignment, i.e., if an aggregated activity requires two aggregated resources, the corresponding detailed activities can only require resources belonging to these aggregated resources. A detailed activity can require more than one resource, but with decreasing probabilities. If the user determines *p* as the probability of requiring only one resource, then the probability of requiring two resources would be *p(1-p)*, three resources *p(1-p)(1-p)*, and so on. The requirements in hours of all the resources that constitute an aggregated resource are added to determine the aggregated requirement. Finally, allowing the testing of the usual heuristic and monetary functions (i.e. VAN) led to the inclusion of cash flows, holding costs and priorities (which are sometimes used to weigh up the penalty costs).

8 Discussion and Conclusions

The computer tool, HierGen, whose functionalities and internal algorithms are presented in this paper, fulfils a gap in the current portfolio of project network generators. It can generate sets of project networks of flat, horizontally aggregated or hierarchical structure, of the desired size and (through the choice of arc saturation) complexity. These project networks can be drawn in a visually structured way. Once generated, they are stored in a database in a defined format, where they can be accessed by the simulation or planning and scheduling tools that use them. Therefore, HierGen or its algorithms can be used to support the simulation of environments in which hierarchical project networks are common, as well as in the testing of hierarchical planning and scheduling techniques.

One of the aims of the tool is to generate project networks that are similar to the project networks found in real life situation. To achieve this aim, users can choose the appropriate Ninitial and Nfinal parameters for flat projects, or choose the number of aggregated activities and of activities in each subnetwork in hierarchical projects.

References

1. Weglarz, J. (ed.): Project Scheduling: Recent Models, Algorithms and Applications. Kluwer Academic, Amsterdam (1999)
2. Demeulemeester, E., Herroelen, W.S.: Project Scheduling: a Research Handbook. Kluwer Academic, Boston (2002)
3. Demeulemeester, E., Dodin, B., Herroelen, W.: A Randon Activity Network Generator. Operations Research 41 (1993) 972–980
4. Kolisch, R., Sprecher, A., Drexl, A.: Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. Management Sci. 41 (1995) 1693–1703
5. Schwindt, C.: Generation of Resource-Constrained Project Scheduling Problems Subject to Temporal Constraints, Report WIOR-543. Universität Karlsruhe, Alemania (1998)

6. Agrawal, M.K., Elmaghraby, S.E., Herroelen, W.S.: DAGEN: A Generator of Testsets for Project Activity Nets. European Journal of Operational Research 90 (1996) 376–382
7. Demeulemeester, E., Vanhoucke, M., Herroelen, W.: RanGen: a Random Network Generator for Activity-On-the-Node Networks. Journal of Scheduling 6 (2003) 17–38
8. Bertrand, J.W.M., Wortmann, J.C., Wijngaard, J.: A Structural and Design Oriented Approach. Elsevier Science, Amsterdam (1990)
9. Bein, W.W., Kamburowski, J., Stallman, M.F.M.: Optimal Reduction of Two-terminal Directed Acyclic Graphs. SIAM J. Comput. 21 (1992) 1112–1129
10. van der Merwe, A.P.: Multi-project Management – Organizational Structure and Control. Int. J. Project Management 15 (4) (1997) 223–233
11. Gordon, V.S., Proth, J.-M., Chu, Ch.: Due Date Assignment and Scheduling: SLK TWK and Other Due Date Assignment Models. Prod. Planning and Control 13 (2) (2002) 117–132
12. de Boer, R.: Resource-Constrained Multi-Project Management – A Hierarchical Decision Support System. Ph.D. Thesis, University of Twente (1998)
13. Motoa, T.-G.: Herramienta Informática de Planificación de Múltiples Proyectos con Enfoque Jerárquico y Análisis de Carga–Capacidad. Ph.D. Thesis, U. Politécnica Madrid (2000)
14. Tormos, P., Barber, F., Lova, A.: An Integration Model for Planning and Scheduling Problems with Constrained Resources. 8th Int. Work. Proj. Manag. and Sched. (2002) 354–358
15. Barták, R., Mecl, R.: Integrating Planning into Production Scheduling: Visopt Shopfloor System. In Kendall, G., Burke, E., Petrovic, S. (eds.): Proceedings of the 1st Multidisciplinary Int. Conf. on Scheduling: Theory and App. (MISTA). Nottingham (2003) 259–278
16. Barták, R.: Modelling Planning and Scheduling Problems with Time and Resources. In: Recent Adv. Computers, Computing Communications. WSEAS, Rethymnon (2002) 104–109
17. Smith, D.E., Frank, J., Jónsson, A.K.: Bridging the Gap between Planning and Scheduling. Know ledge Engineering Review 15 (1) (2000) 61–94
18. Hax, A.C., Meal, H.C.: Hierarchical Integration of Production Planning and Scheduling. In: Geisler, M.A. (ed.): Studies in the Management Science, Logistics, Vol. 1. North-Holland-American Elsevier, New York (1975)
19. Bitran, G.R., Tirupati, D.: Hierarchical Production Planning. In:Graves, S.C., Rinnooy, A. H.G., Zipkin, P.H. (eds.): Logistics of Production and Inventory. Handbooks in OR and Manufacturing Science, Vol. 4, North-Holland (1993)
20. Mehra, M., Minis, I., Proth, J.M.: Hierarchical Production Planning for Complex Manufacturing Systems. Advances in Engineering Software 26 (1996) 209–218
21. Vicens, E., Alemany, M.E., Andrés, C., Guarch, J.J.: A Design and Application Methodology for Hierarchical Production Planning Decision Support Systems in an Enterprise Integration Context. Int. J. Production Economics 74 (2001) 5–20
22. Baptiste, P., Le Pape, C., Nuijten, W.: Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems. Kluwer Academic, Boston (2001)
23. Cesta, A., Oddi, A., Smith, S.F.: A Constraint-Based Method for Project Scheduling with Time Windows. Journal of Heuristics 8 (1) (2002) 109–136

Macroscopic Treatment to Polymorphic E-mail Based Viruses*

Cholmin Kim[1], Soung-uck Lee[2], and Manpyo Hong[1]

[1] Internet Immune System Laboratory, Graduated School of Information and Communication, Ajou University
{ily,mphong}@ajou.ac.kr
[2] Shingu College
suleeip@yahoo.co.kr

Abstract. Today's E-mail based viruses are proficient to spread and possess polymorphic facilities to make them difficult to detect. The results of previous researches to treat these viruses are not very different from detection methods for ordinary host based viruses. These methods are weak to detect a polymorphic E-mail based virus. Thus, to deal successfully with the limitations of previous works, the research of this paper suggests the idea, which can detect polymorphic viruses from the macroscopic behavior of the infected mail. In this paper we will show that the spread of viruses can be contained and the number of infected clients can also be decreased at the equilibrium state by advancing some portion of E-mail servers.

1 Introduction

E-mail based viruses, or internet worms, are one of the most serious mischief to internet users. Current viruses can be run automatically and can consume most of the bandwidth of a network by continuously sending bulk of E-mails. Because of this ability current E-mail viruses are as dangerous as a DDoS attack.

The results of previous researches to treat the E-mail based viruses are not very different from detection method for ordinary host based viruses [1]. In other words, scanning the virus signature is the major job for a recent anti-virus product. These methods are weak to detect a polymorphic virus [2, 3].

Many researchers try to find a good detection method for polymorphic virus. Then the virus becomes more complex to make it harder to decide whether it is a polymorph of some sort or not. To overcome this shortcoming, some researches suggest heuristics, but they still fails to notice a special behavior of E-mail based viruses [3, 5].

Thus we suggest some new detection methods that can detect polymorphic E-mail based viruses using the macroscopic behavior of virus. We will show that our method is effective to contain the spread of viruses. Some simulation results for the viral spread in macroscopic model will be shown.

* This work was supported by grant No. (R05-2003-000-11235-0) from the Basic Research Program of the Korea Science & Engineering Foundation.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 867–876, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The rest of this paper is organized as follows: In Section 2, we introduce the brief overview of macroscopic researches of a virus spread, our previous work and virus polymorphism. Section 3, we suggest the idea how to detect and treat polymorphic viruses in a macroscopic view. Section 4 describes the simulation for our idea and its result. Finally we summarize our findings in Section 5.

2 Background and Related Works

2.1 Macroscopic Approach

The suggested idea in this paper is based on macroscopic approach. This macroscopic research to viruses was started from the researches of how viruses can spread. The epidemiology also can be addressable to computer viruses [6, 7]. In other words, if we define "vectors" like in biological diseases, we can clarify the characteristic of the spread of virus between these vectors. For example, E-mail can be a vector of a virus. This virus will have a special characteristic that depends on the characteristic of a E-mail. But, because it is a virus, it will have some odd behavior contrary to normal mail. Thus macroscopic treatment to this type of virus is to find an abnormal mail delivery flow which can be doubted as a suspicious mail delivery.

Macroscopic approach also emphasizes that if we define the vector of a virus mathematically, we can depict the viral behavior of its spread [8]. For instance, if the viral environment consist of fully connected graph, it is easy to depict the spread of virus. Two variables are introduced, the viral birth rate β and death rate δ. These two symbols have the same meaning with the ones in stochastic process. The viral birth rate is the likelihood that the neighbor of infected nodes can be infected and the death rate is the probability that the virus will be eliminated from the environment.

This models, however, cannot exactly depict the real virus behavior, because, it is hard to get the real distributions for each node connectivity in computer networks. Thus some assumptions are needed to simulate the spread of viruses [9, 10].

2.2 Our Previous Research

Our previous research of this paper was how we can separate virus-attached mail from normal ones [2]. If we compare the virus-attached mail with a normal one, we can find that it is delivered to many destinations from a single host. The reason for this phenomenon is that the virus reads the address book of the host and sends self-attached mails to every client in the address book. However, there can be public announcement to every member of a group. In that case, same mail will be delivered to every member of that group for the normal purpose. Therefore, to avoid a wrong decision, we decide a mail has a virus if it is about to be delivered from a single host to multiple destinations after once it has been delivered in the same manner unchanged.

The E-mail server is located at the best place to monitor the behavior of the virus. In the Ingress Filtering, used in IDS to prevent DDOS attack, the best place to monitoring the packet flow is vertex cover [11]. In E-mail delivering infrastructure, two mail servers, sending-side server and receiving-side server, must be engaged in

Phase 1: If the sending-side mail server detects bulk of mails from single source then it attaches a tag to those mails to identify suspicious ones.
Phase 2: If a mail server receives a tagged mail then it maintains this mail as signature of virus.
Phase 3: If receiving-side mail server detects bulk of mails from a single source and the mail is same with some signature, this server decides it as a virus.

Fig. 1. The virus detection mechanism in previous research

delivering mail to their clients. In order to achieve the detection effectively, we have advancing E-mail servers to possess the ability of 3-phases observation in Fig. 1.

The advanced server in the previous idea can keep the virus from spreading by not delivering the mails, which have been decided to have virus in Phase 3. In addition to this advantage, the server can report the suspicious mail to the anti-virus company to contribute to produce a signature for anti-virus. This anti-virus system can detect the virus more precisely.

2.3 Virus Polymorphism

Recent viruses use polymorphic technique to evade from the anti-viruses. The trend of the technique becomes more and more complex. Frequently used techniques are encryption and code swapping [5]. In the strict sense of the word, signature based anti-viruses are unable to detect these kind of polymorphic viruses [3]. Same polymorphic techniques are used in E-mail based viruses. It means that neither general anti-virus nor our previous work can detect polymorphic E-mail based viruses. The previous work compares the body of pre-saved suspicious mail with newly detected suspicious one (**Phase 3** of Fig. 1.). The system must do this step to avoid the false positive case. Thus our previous method is not perfect and a new method that does not require the signature matching is needed.

3 The Macroscopic Treatment to E-mail Based Viruses

3.1 The Idea

The basic idea of the work is simple. Since one of the important objectives of a virus is spreading its descendent as quickly as possible, E-mail based viruses tend to send bulk of self-attached mail. Then the next infected host will show the same behavior if the delivered virus has been activated. Thus, as described in section 2.2, it shows some odd mail flow. Bulk of mails is delivered from a single host to multiple destinations continuously (we call it one-to-many chain).

We direct our attention to how many recipients of bulk mails send bulk mails again in limited time interval. In other words our system will monitor the bulk mails and its recipients. If a portion of recipients send bulk mails again within some interval, we suspect that the first bulk mail, and its decedent mails, has virus. By this idea, we expand our previous work into the 4 phases in Fig. 2.

Phase 1: If the sending-side mail server **S** detects bulk of mails from a single source, it builds a recipient list and attaches a tag to those mails to identify suspicious ones.
Phase 2: If the receiving-side mail server **R** receives a tagged mail, it maintains the address of **S** and mail recipient.
Phase 3: If **R** detects bulk of mails from a single source and the mail sender was a bulk mail recipient, **R** notifies it to **S**. **S** updates the recipients list with notified recipient checked.
Phase 4: If the portion of checked recipients, in the bulk mail recipients list, run over some threshold, **S** decides that the bulk mail in **Phase 1** was containing the virus.

Fig. 2. The virus detection phases in suggested idea.

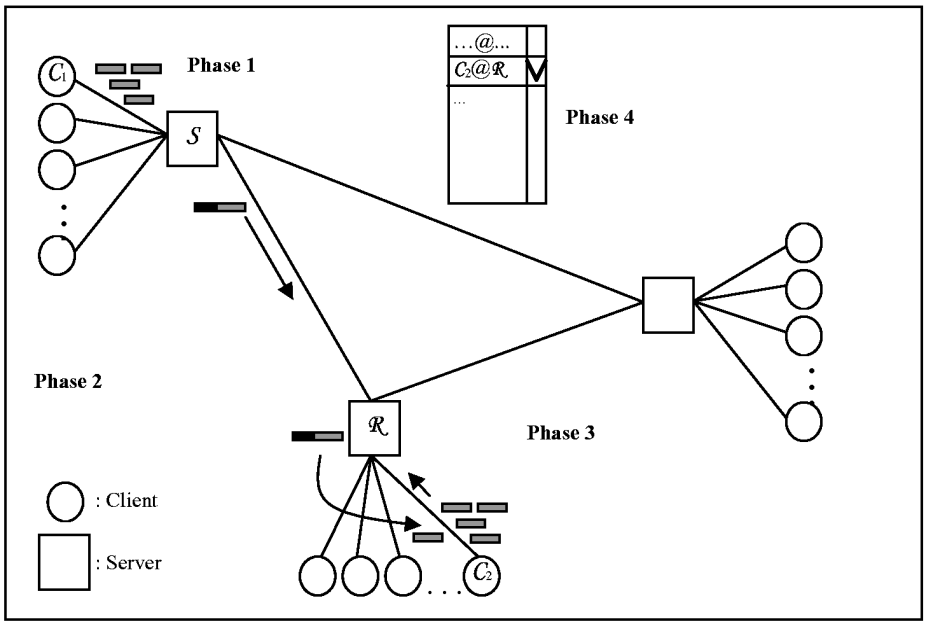


Fig. 3. The scenario of virus spread and detection.

Fig. 3. describes the scenario of virus detection based on this idea. Let's assume that the E-mail client $\mathbf{C}_1$ has been infected by a virus, and the address book of $\mathbf{C}_1$ contains the address of $\mathbf{C}_2$. Now we can explain the situation of Fig. 3 in 7 steps.

***Step* 1**. The virus in $\mathbf{C}_1$ is beginning to activate. It replicates itself to mails and sends these mails to every client in the address book of $\mathbf{C}_1$.
***Step* 2**. **[Phase 1] S**, the E-mail server of $\mathbf{C}_1$, detects bulk of mails from $\mathbf{C}_1$. It builds a recipients list for these mails and attaches a tag to each mail before being delivered. Tagged mails are regarded as suspicious.

***Step* 3**. **[Phase 2] R**, the E-mail server of $\mathbf{C}_2$, receives one of the tagged mails. It preserves the address of **S** and recipient. The mail is delivered to $\mathbf{C}_2$ after eliminating the tag to make the mail same with the original one.
***Step* 4**. Now the virus infects $\mathbf{C}_2$ and beginning to activate. It replicates itself to mails and sends these mails to every client in the address book of $\mathbf{C}_2$.
***Step* 5**. **[Phase 3] R** detects bulk of mails from $\mathbf{C}_2$. It compares the address of $\mathbf{C}_2$ with the recipient address that has been preserved in ***Step* 3**. If these two addresses are matched, **R** notifies it to the **S**. **S** updates the recipients list with notified recipient checked.
Step 6. Some other servers, which received the virus attached mail from $\mathbf{C}_1$, will do same steps from ***Step* 2** to ***Step 5***.
***Step* 7**. **[Phase 4]** The portion of recipients checked, in the bulk mail recipients list of **S**, run over some threshold, **S** decides the bulk mails in **Phase 1** had been containing the virus.

After that, the advanced E-mail server in the suggested idea can prevent the virus from spreading by do not delivering the mail from the senders who have been decided to receive the virus attached mail. This server also can report the suspicious mail to the anti-virus company to update a signature DB of anti-virus engine in the same manner as our previous work.

4 Simulation

In the simulation, we virtually construct our own E-mail world, and to place our work There are no layers in E-mail server system.

1. There are groups of E-mail clients. A client is more likely to have addresses of other client in the same group than the client of other groups.
2. The address book of a client can not be changed during experiment.
3. Every client confirms his mail once per day from his mail server.
4. Some E-mail servers have advanced facilities.
5. Some error, which is caused by deciding a normal mail as an evil one, or false positive error, is negligible.

The meaning of the first assumption is that every client has only one server on its way to deliver his mail to the receiving side server (but, in fact, there is some E-mail system which has multiple layers of servers). The second assumption represents real conditions of E-mail clients. Large companies or public organizations use its own E-mail server and every client of that server is likely to be a member of those companies or organizations. So the client is likely to know the address of other clients of that server. By the fourth assumption, we can define the grain of the clock of our experiment as a day. Because, in programming detail to model the spread of virus from host to host, we've not used real time interrupt but used loop and polling technique which asks to the client if he has virus or if the virus is activated. In this context, one execution of the loop can be considered as an elapse of one day. By the last assumption, we can ignore the false positive case which was mentioned at Section 2.2 (In fact this case is negligible in real world).

***Birth-rate*:** The probability that a virus will be executed by a user or by some accident in a host which has virus.
***Death-rate*:** The probability that a virus will be eliminated by anti-virus program. The host require proper anti-virus for the virus before applying this probability.

Fig. 4. Definitions of Birth and Death-rate

```
1 if client has virus mail
2 {
3 if virus had been reported & anti-virus was propagated
4 filter by death-rate
5 {
6 filter by birth-rate
7 {
8       set virus run
9       loop for each client in address book of this client
10      {
11      set has virus mail
12      }
13      }
14      if virus hadn't run
15      {
16      unset client has virus mail
17      }
18 }
19}
```

Fig. 5. Algorithm for experimental model

Before the experiment in this environment, we need to define some concepts. In the previous Kephart's work, which was to simulate the spread of viruses, the experiment had been done in given topology with virus birth-rate and death-rate. The experiment of our work use these terms too. We have defined those in our words. Fig. 4 is our definitions.

The experimental model obeys the algorithm of Fig. 5. The algorithm is the body of the loop for each client. If the loop was executed for all clients then it means that one stage of the experiment has been done.

The shaded area of Fig. 5 is the part of adopting the death-rate. We have done both experiments with this part and without it on purpose.

The meaning of not adopting death-rate is that the effect of anti-virus is ignored. In contrast, the meaning of adopting death-rate is that the effect of anti-virus is considered. Because of its accurate detection and elimination for the known viruses, the spread of virus also be contained by the appearance of the anti-virus, which can use signature reported by advanced server. It requires some delay to propagate anti-virus to reasonable number of users. It is reflected in line number 3. From line number 5 to 18 is the case which passes the death-rate. It means that the anti-virus has failed to detect the virus or the anti-virus is not circulated yet. Then the birth-rate of virus is applied. If the death-rate was applied to the client then the client may not have the virus and the birth-rate is meaningless to this client. By the birth-rate, some portion of

ADDR_BOOK_SIZE : Size of Address book for each client. We use a uniform distribution for this value per each client.
MAX_INNER_CONNECTION : Number of mail addresses of clients who are in the same group.
MAX_INTER_CONNECTION : Number of mail addresses of client who are in the other group.
MAX_CLI_NUM_PER_SERVER : Max available number of clients per each E-mail server.
MAX_CLI_NUM : Total number of clients in our E-mail world.
MAX_SERVER_NUM : Total number of servers in our E-mail world.
ADV_SERVER_PORTION : The portion of advanced server to whole server.
ANTI-VIRUS_PROPAGATION_DELAY : Required stages to propagate anti-virus to each clients.

Fig. 6. Definitions of some parameters.

clients, which has virus, are infected to virus and they will propagate the virus to the other clients registered in their address book, from line number 7 to 13 reflects it. From line number 14 to 17 is the case that the birth-rate was failed to apply. The virus is ignored and assumed that can not be activated anymore in the host.

We use *SIMSCRIPT II.5*, which is a script based simulation tool, for experiments [12].

In addition to two stochastic process rates, in programming detail, we use several parameters to make our simulation more reasonable. Fig. 6 shows those parameters.

In the first assumption of this section, we've mentioned a group of E-mail clients. By this assumption and simulation, we realize that **MAX_INNER_CONNECTION** can affect the speed of viral spread like the first description for E-mail group in the beginning of this section. Ordinary people have more addresses for clients in the same group.

MAX_INTER_CONNECTION depends on the value of **MAX_INNER_ CONNECTION**. Because, if we add these two values, it will be the whole address book size.

We show the result of the experiments. As mentioned in section 2.1, the shape of the graphs is dependent on the birth-rate and death-rate. Due to the definitions of birth-rate (β) and the death-rate (δ) we can derive the fraction of infected nodes, $a(t)$, in random graph as a function of time [8]. In our experiments, 'a' denotes the fraction of infected clients.

$$da / dt = \beta a(1 - a) - \delta a \tag{1}$$

If we solve this simple nonlinear differential equation, we can get $a(t)$.

$$a(t) = [a_0(1 - \rho)]/[\, a_0 + (1 - \rho - a_0)e^{-(\quad)t}] \text{ when } \rho = \delta / \beta \text{ and } a_0 = a(t = 0) \tag{2}$$

The result of our works are very resemble with *a(t),* because n*,* the address book size in our case, is sufficiently large. X-axis of each graph denotes the number of stages in the experiment and Y-axis denotes the number of infected clients. In every experiment there are 10,000 clients in virtual E-mail world, we fix **MAX_CLI_NUM** to 10,000. Fig. 7, 8 and 9 show the experiments that have been done without the death-rate. Fig. 10 shows the experiment with the death-rate.

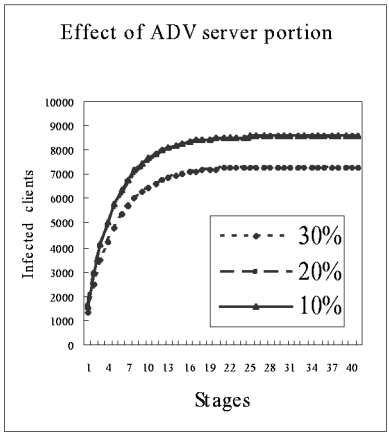


Fig. 7. The effect of ADV server portion

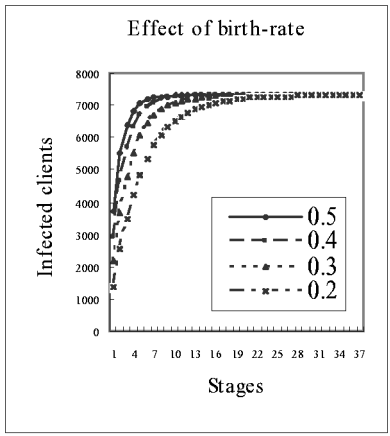


Fig. 8. The effect of the Birth-rate

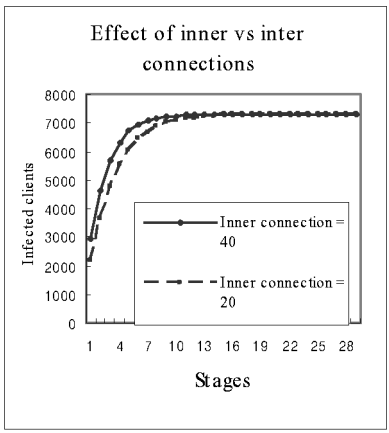


Fig. 9. The effect of inner vs. inter connection

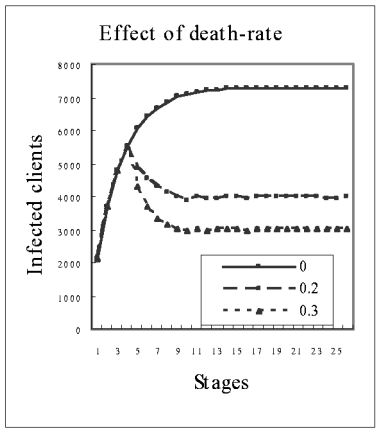


Fig. 10. The effect of the Birth-rate

In experiments of Fig. 7, the birth-rate is fixed to 0.3. Other parameters are fixed too, except the portion of advanced servers. By Fig. 7, we can see that the spread of virus is contained and the number of infected clients is decreased by the suggested idea. If we see the graphs horizontally, as the portion of advanced server grows, we can see that the time delay, needed for virus spread, is become larger and larger. In other words, the graph which denote higher portion of advanced server situation is more leaned to the right side than the graph of lower advanced server portion. On the other hand, if we see the graphs vertically, we can see that the number of infected clients decreases as the portion of advanced server increases.

Fig. 8 shows the effect of birth-rate. As the birth-rate go higher, the viral spread becomes faster and faster. On the contrary the number of infected clients at the equilibrium state is not changed. Fig. 9 shows the effect of the number of inner and inter connections. Since the number of inter connection is related with the probability

of global infection, it affects the speed of viral spread. However, it can not affect the equilibrium state either.

Fig. 10 shows the effect of the death rate. If the signature for the current generation of virus can be developed then the Anti-Virus system using signature can eliminate the virus more accurately. Consequently, the number of infected clients is decreased. In the death-rate case, however, we must assume that the signature for the every generation is developed.

5 Conclusion

In this paper we have describes the idea to treat polymorphic E-mail based viruses in macroscopic view. We have focused our interest on the characteristics of vector of these viruses. When, that is, the virus tends to spread self replicated ones to victim hosts, it delivers bulk of self-replicated mails by searching the address book of the infected one. We used this phenomenon to detect suspicious E-mail. We modified some portion of the E-mail server and applied 4-phases detection policy to each advanced servers. Because these servers, advanced ones, could be the articulation point of virus, we could held down both The speed of viral spread and the number of infected clients at the equilibrium state. Since our idea do not concern with the signature, it can detect the polymorphic virus. Finally, we suggested the result of simulation for our idea. We showed that what parameter can influence the spread of virus.

By the method in this paper we can treat unknown polymorphic viruses. Currently used heuristic engines are too heavy to detect unknown polymorphic virus accurately and need to have too much computing power. If the suggested idea is used for preprocessing of these heuristic engines, it can contribute to detecting new viruses quickly and accurately. It can also be used for suspicious mail reporter for signature based anti-virus engines.

References

1. Igor Muttik, STRIPPING DOWN AN AV ENGINE, Proceeding of the *VIRUS BULLETIN CONFERENCE,* (2000) 59–68.
2. Cholmin Kim, Seong-uck Lee, Hyeongchol Jung, Yoosuk Jung and Manpyo Hong, Macroscopic Treatme-nt to E-mail Based Viruses, Proceeding of the *SAM'02,* (2002)
3. Gabor Szappanos, ARE THERE ANY POLYMORPHIC MACRO VIRUSES AT ALL? (…AND WHAT TO DO WITH THEM), Proceeding of the *VIRUS BULLETIN CONFERENCE,* (2002)
4. Computer Associate International Inc., CA Releases Top 10 Virus List for 2003, http://www3.ca.com/ press/pressrelease.asp?id=1856, (2003)
5. Vesselin Bontchev, MACRO AND SCRIPT VIRUS POLYMORPHISM, Proceeding of the *VIRUS BUL-LETIN CONFERENCE,* (2002) 406–438
6. Jeffrey O. Kephart, David M. Chess and Steve R. White, Computers and Epidemiology, *IEEE Spectrum*, v30 n5, (1993) 20–26
7. Frederick B. Cohen.: A Short Course on Computer Viruses 2nd Edition, *John Wiley & Sons, Inc,* (1994) 121–134

8. Jeffrey O. Kephart, How Topology Affects Population Dynamics, Proceeding of the *Artificial Life III Studies in the Science of Complexity Proc.* Vol. XVII, (1993) 447–463
9. Winfried Gleissner, A mathematical theory for the spread of computer viruses, *Computers & Security,* vol. 8, (1989) 35–41
10. Tipet, The Tipet Theory of Computer Virus Propagation, *Foundationware*, USA, (1990)
11. Kihong Park, Heejo Lee, On the Effectiveness of Route-Based Packet Internets Filtering for Distributed DoS Attack Prevention in Power-Law, Proceeding of the *ACM SIGCOMM,* (2001)
12. Edward C. Russell.: Building Simulation Models with SIMSCRIPT II.5, *CACI Products Company,* (1989)

Making Discrete Games*

Inmaculada García and Ramón Mollá

Computer Graphics Section
Dep. of Computation and Computer Systems
Technical University of Valencia
{ingarcia,rmolla}@dsic.upv.es

Abstract. Current computer games follow a scheme of continuous simulation, coupling the rendering phase and the simulation phase. That way of operation has disadvantages that can be avoided using a discrete event simulator as a game kernel. Discrete simulation supports also continuous simulation and allows rendering and simulation phase independence. The videogames objects behavior and interconnection is modeled by message passing. Discrete games require lower computer power while maintaining the videogame quality. This means that videogame may be run in slower computers or the game quality may be improved (artificial intelligence, collision detection accuracy, increase realism).

1 Introduction

Many open source free computer games lack from internal organization, employ rudimentary simulation techniques and are not representative of current technologies [2]. Most of videogames kernels studied are in practice only rendering kernels (3D GameStudio [3], Crystal Space [5] or Genesis 3D [9]). Those engines follow the same simulation scheme as the conventional open source commercial videogames (DOOM v1.1 [10], QUAKE v2.3 [14], Serious Sam [4] or the Fly3D kernel v2.0 [17][18]).

We have selected in the present study the videogame kernel Fly3D because it is a C++ highly structured and widely documented open source standalone code. The simulation phase and rendering phase are coupled and separated. Its SDK includes tools to create games easily.

1.1 Disadvantages of Videogame Continuous Simulation

Simulation techniques [1] used in videogames suppose in many cases to consider computer games as continuous systems [11]. Implementing computer games as continuous coupled systems have disadvantages over discrete schemes.

Inefficient Simulation Scheme
All objects in the scene graph are simulated, although many objects will never generate events. Some videogames allow to access only to active objects list. Mainly,

* This work has been funded by the MCYT TIC2002-04166-C03-01.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 877–885, 2004.
© Springer-Verlag Berlin Heidelberg 2004

once the objects are accessed to be rendered, they are also simulated. A new simulation cycle requires always an entire world rendering since simulation phase and rendering phase are coupled. Test [16] made on current personal computers and commercial videogames show that more than 70% rendering power may be wasted if the system loop overpasses the Screen Refresh Rate (SRR).

Erroneous Simulation
The objects priority for simulation depends on the objects position in the scene graph. For this reason, events are not time ordered. The simulation may be erroneous because of disorderly events execution or even the execution of cancelled events.

Sampling Frequency
The videogame objects sampling frequency is the same for all objects, independently of their requirements. If objects behaviors do not match Nyquist-Shannon theorem, they will not be simulated properly, loosing events, not detecting collisions,...The objects will be undersampled. Objects with a very slow behavior may be oversampled, wasting computer power. The system is not sensitive to times lower than the sampling period. The simulation events are artificially synchronized to match the world sampling period. They are not executed in the very moment when they happen. The sampling frequency depends on topics that can change during the game, such as available computer power, world complexity, other active tasks in system, network overload or current simulation and rendering load. So, the sampling frequency is variable, not predefined and not configurable.

Quality of Service
Once the videogame has been designed, there must be defined the videogame objects QoS (Quality of Service). The meaning of QoS depends on the given object. For instance, the Render object QoS depends on general rendering aspects as SRR (fps), anti-aliasing (FSAA), motion blur, accumulation buffers or screen resolution. Common videogame objects QoS depends on topics such as sampling frequency, collision detection, artificial intelligence (AI), textures, object geometry, lighting or lightmaps. The programmer defines the other videogame objects QoS: object geometry, polygons number, object size, texture color depth or textures total number. Programmers traditionally allow the user to define some videogame rendering characteristics as anti-aliasing or screen resolution. This is a way to adjust the render object QoS to the computer power where the videogame is executed. In a continuous videogame the SRR can not be defined by the programmer or by the user. It is hardly dependent on the system load. If the system load grows, the SRR falls.

1.2 Videogame Improvement

Videogames objects (avatars, missiles or guns) can mix both continuous and discrete behaviors. So, videogames are actually hybrid systems [13]. Discrete simulation paradigm has advantages over continuous simulation since only those objects that change their state produce events and consume computer power.

- The object priority in simulation depends on the time its events are set. Priority does not depend on anything more. Events are strictly time ordered executed. The execution may be slower, but the game is correctly simulated.
- Every object has its own private sampling period independent of every other system object. No over or undersampling appears if there is enough computer power. There is no restriction in the sampling frequencies. They are constant during the whole game execution. They can be managed by the program explicitly.
- Events happen in the very moment they were planned and they are not artificially synchronized.
- The discrete simulation scheme supports continuous simulation also. The opposite is not possible.
- Discrete videogames allows the independence of the SRR and the system load in normal conditions. Besides the SRR can be defined and/or adjusted during the videogame execution automatically or explicitly by the programmer. So, the QoS can be defined for each object or for the whole system.
- If simulation and rendering are decoupled, scenes are rendered more quickly even when the higher-level animation computations become complex [15]. This decoupling increases system performance [6].

1.3 Discrete Event Simulators

The test made in this paper uses the discrete event simulator DESK [8] to be integrated into Fly3D to perform the application simulation to change the simulation scheme from continuous to discrete. DESK is a C++ open source code simulator like Fly3D. The DESK capabilities have been adapted to those required to support videogames becoming into Game DESK (GDESK) [7]:

- DESK is not a real time simulator. An event in GDESK only happens if the event time stamp is reached or exceeded by the real system time.
- An event models the communication between two objects (or an object to itself). It is produced by a videogame object and it always has a destiny object. The events parameters are partially defined by the videogames programmer.
- There are two ways to finish a simulation using DESK: to determine a maximum simulation time or to simulate until the events queue is empty. Neither of them is appropriated to be used to control the videogame simulation. A videogame only finish if the user explicitly generates a finish event. In a videogame, if the events queue is empty, the game is paused until a new user or network event is generated.

2 Objectives

We want to study if the performance is increased when:

1. Changing the videogame simulation paradigm from continuous to discrete.
2. Decoupling the simulation phase and the rendering phase.

To achieve those objectives we have compared the same game kernel implemented in both paradigms. We have compared the Fly3D continuous game kernel to the same kernel using GDESK as the simulation kernel Discrete Fly3D (DFly3D).

3 Tool Description

Fly3D main loop follows a typical scheme of continuous simulation that couples the simulation phase and rendering phase [12]. The simulation takes care of the time elapsed from last simulation until now. The simulation process calls each active object simulation function. A complete simulation and rendering is done for each main loop step.

GDESK is a real time applications simulation kernel that copes with the videogame events (messages) handling. GDESK controls the objects communication by message passing and maintain the events ordered by time until their time stamp is exceeded. Any videogame element must be a GDESK object. GDESK treats any object in the same way. Messages are the passive elements that communicate objects. The system dynamic is modeled by message passing. GDESK object inherits from the GDESK basic object class. The GDESK basic object contains functions to send a message to another object (or to itself) and to receive a message. It is a virtual function fully defined by the programmer that implements the object response to an incoming message. This is the simulation object function.

The message interchange process is controlled by GDESK. It catches the messages sent from one object to another and stores them ordered by time. This process is transparent to the objects in the system. The message time is converted to an absolute videogame time using the GDESK simulation clock. The message is stored ordered by the absolute time. Once a message is stored, GDESK goes on working, testing if the first stored message time stamp has been reached. If it is so, GDESK pop-ups the message and it send the message to the destiny object.

3.1 DFly3D: Integrating GDESK into Fly3D

DFly3D is the modified Fly3D kernel result of using GDESK to manage Fly3D events. Integrating GDESK into Fly3D, every videogame event is managed by GDESK. GDESK does not interfere with other game components and does not imply to change the structure of the scenes description files or characters. It only modifies the system events management and the way the simulation is made. DFly3D matches GDESK basic object methods specific from Fly3D.

There are two object types in the DFly3D kernel:

- System objects: they are the objects explicitly created in the GDESK integration to develop some Fly3D kernel tasks. They develop Fly3D engine tasks such as rendering or console events management
- Videogame objects: they are the objects created by the videogame programmer (walls, characters or weapons).

Both object types generate messages because they are GDESK basic objects. These messages are managed by GDESK in the same way. Videogame objects join the Fly3D basic objects characteristics with GDESK basic objects characteristics.

While Fly3D main loop follows the typical scheme of coupled continuous simulation, the DFly3D main loop supposes to change the Fly3D simulation function by the GDESK simulation function and remove the rendering from the main loop (figure 1). The rendering is integrated in the system as any other game object. The main loop copes with discrete events while decoupling the system.

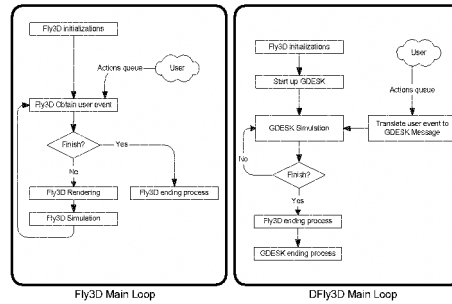


Fig. 1. DFly3D and Fly3D main loops

DFly3D Rendering Process
DFly3D allows the simulation phase and rendering phase decoupling. The rendering process is controlled only by the render system object. The rendering process starts using a message sent by the initialization game routine before the simulation starts. The render object must decide the very moment to generate a new render event, that is, the time T_R when the new frame $N+1$ is calculated. Then it sends an event to itself to happen in T_R. The render object has to generate only as many renders as the SRR in order to avoid rendering frames that will never be shown on the screen. It has to adapt the number of renderings to the system load, assuring the QoS set by the user or the programmer (minimum number of renderings per time unit).

4 Results

4.1 Test Conditions

The results have been obtained creating a videogame consisting on some balls jumping and colliding. Two videogame versions have been implemented for both Fly3D and DFly3D kernels. The balls number for tests has been changed increasing both simulation and rendering load. Both systems have been tested in a commodity PC Pentium 4 (2 GHz) with 512Mb Ram with the Nvidia GeForce 4 MX440 graphic card. The videogame topics have not been changed for both the continuous and the discrete system. The scene complexity, files size, management internal structures, graphic resolution, structure of the scenes description files or characters have been maintained. The only difference between Fly3D and DFly3D is the system events management. The resources consumed for both systems are similar, but the CPU load for each system is different depending on the simulation scheme used in the videogame programming. The following tests try to evaluate this aspect.
Let it be:

- SSF: system sampling frequency.
- OSF_i: object *i* sampling frequency that models the object *i* behavior.
- $OSF_{i\,min}$: minimum OSF_i to simulate the object i properly.
- $OSF_{i\,max}$: maximum OSF_i to simulate the object i.
- SRR_G: SRR produced by the videogame.
- SO: Screen Oversampling. SRR_G never seen on the screen per time unit.

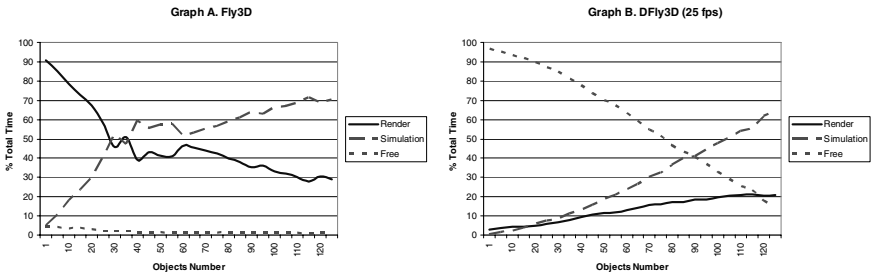


Fig. 2. DFly3D and Fly3D time use (percent)

4.2 Continuous System and Discrete System Comparison

We say that the computer system is *collapsed* when a system is not able to show the scene properly (incorrect objects behavior or low SRR). A system could be collapsed due to the videogame scene or simulation complexity or because of the low computing power. If the system is collapsed both kernels do not allow running the videogame properly. The continuous kernel typically undersamples the objects making their movements chaotic and collision detection fails many times (equation 1). The discrete kernel produces a correct output but the system evolution is slower (equation 2). The given moment of system collapse depends on the videogame complexity and the kernel used.

$$\exists i \,/\, OSF_{i\,min} > SSF\ . \tag{1}$$

$$\forall i\ OSF_i > OSF_{i\,min}\ . \tag{2}$$

The following comparison is based in a not-collapsed system.

1. SRR

– Continuous systems. The whole continuous system has a common SSF (equation 3). If an object needs to be sampled OSFi, the object i may be undersampled (equation 4) and the object behavior can be erroneous (not detected collisions or erroneous object trajectories on the screen).

– Discrete system. SSF does not exist in discrete systems because each object i has its own OSFi. The render object maintains the sampling frequency under SRR (equation 5). The SRRG value is configurable and it can be adjusted by the system.

$$SRR_G = OSF_i = OSF_j = SSF\ .\ \forall \text{ Videogame objects } i, j \,/\, i \neq j \tag{3}$$

$$\exists i \,/\, OSF_i > SSF\ . \tag{4}$$

$$SRR \geq OSF_{RENDER} = SRR_G. \tag{5}$$

2. Computer power

– Continuous systems. They waste computer power calculating unnecessary renderings (equation 6).

– Discrete system. The saved time can be used in the discrete system to adapt the OSF_i (equation 7), trying to maintain $OSF_{i\,max}$ or to achieve at least $OSF_{i\,min}$.

$$SO = SRR_G - SRR = SSF - SRR \,. \tag{6}$$

$$OSF_{i\,max} > OSF_i > OSF_{i\,min} \,. \tag{7}$$

3. Free time

– Continuous system (figure 2.A). The application time is shared by the simulation and the rendering processes. The system uses nearly the 100% of the time rendering and simulating. For each main loop step each object is simulated once and the scene is rendered. There is no free time and no free resources.

– Discrete system (figure 2.B). The rendering process and the simulation process are not dependent (decoupling). So, the videogame time is not shared by rendering and simulation. The system uses only the 100% of CPU time if the system is collapsed (the system load is bigger than the computer power). The discrete system always freed more time than the continuous system. The resources consumed by a discrete videogame are always lower than the resources consumed by a continuous videogame.

4. System load

– Continuous system. The system uses nearly the 100% of application time regardless of the system load. An increase in the simulation load supposes to decrease the rendering time (SRR decreases).

– Discrete system. An increase in the system load decreases the free time. Only if the system is collapsed, an increase in the system load decreases the SRR and the system slows down homogenously among every object in the system.

5. Real time

– Continuous system. The system always follows the real time changing SSF.

– Discrete system. If the system is collapsed the simulation time is behind the real time because the computer power is not enough to simulate and render matching the real time. If the system is not collapsed, it executes the events at real time. When the simulation time overpasses the real time, the simulator is stopped until the real time is achieved (synchronization with the real time). This situation is produced when the next event time to be happen later than the current real time.

6. Frame rate

The tests showed that the maximum frame rate that is able to generate a discrete simulator is always a bit bigger than the one generated by a continuous simulator (figure 4.A). The OSF in a discrete system uses to be lower than the OSF in a continuous system when the system support high load (equation 8). The OSF in a discrete system is not dependent of the SRR. The overload produced by the events management in discrete system is not significant (under 0.001%). When the maximum frame rate is achieved, both systems use the 100% of CPU time.

$$SSF > OSF_i \,. \; \forall \text{ Videogame object } i \tag{8}$$

7. OSF

– Continuous system. The OSF cannot be defined. It is the same for all objects, including the render object (equation 3).

– Discrete system. The OSF can be fixed by the programmer or by the user. The OSF is configurable and it can be explicitly adjusted during videogame execution automatically or by the programmer. Each videogame object has its own OSF. The videogame characteristics that depend on the OSF are object position, collision detection, AI,…

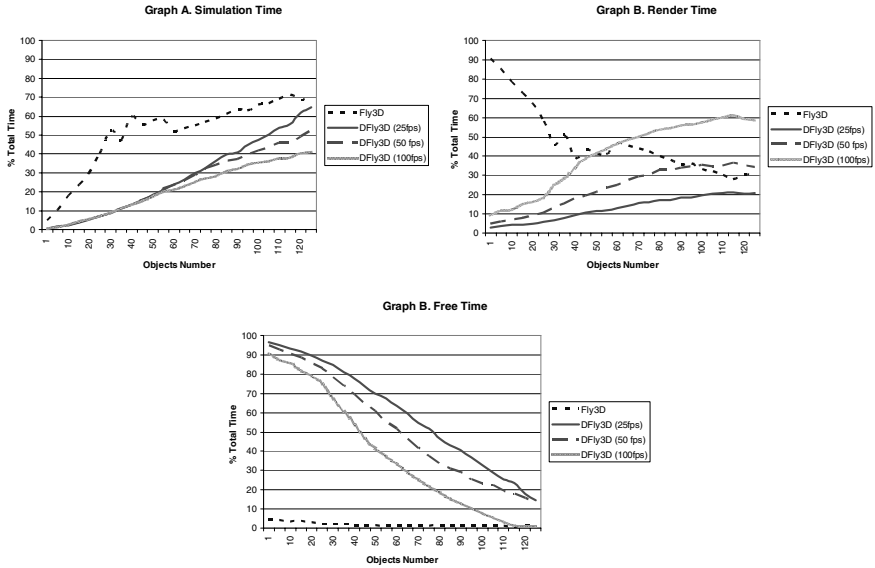


Fig. 3. DFly3D and Fly3D times comparison

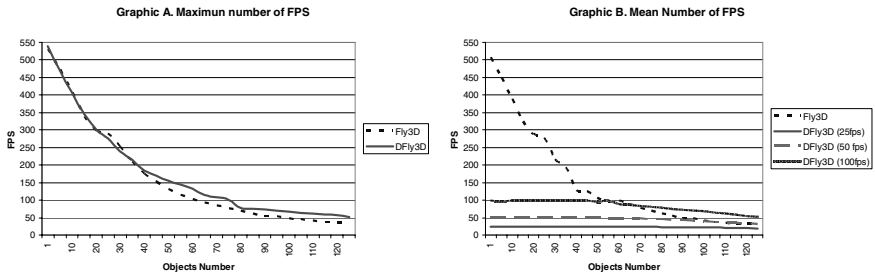


Fig. 4. DFly3D and Fly3D fps

Figure 3 shows the total execution time ratio that is used by both kernels: rendering, simulating and the free time. The DFly3D kernel results are obtained generating 25, 50 and 100 fps. In these test, the continuous system consumes an outstandingly bigger simulation time than the discrete system (figure 3.A) if the videogame complexity is not too high because of the continuous system oversampling. This is also true for the rendering phase (figure 3.B). In the discrete system, there is not oversampling, so those times are released (figure 3.C). The sampling frequency is maintained to the levels defined by the programmer. The freed time in a discrete system is always bigger than the freed time in a continuous system.

Figure 4 shows the number of fps generated by both systems. Both systems achieve a similar maximum number of fps (figure 4.A). The current commercial graphic devices do not need a SRR so high. The discrete system allows to define the SRR generated while the system is not collapsed. The discrete videogame may adjust the SRR to the graphic output device (figure 4.B) avoiding unnecessary renderings.

5 Conclusions

Although continuous simulation games have been the main stream during the last years, this paradigm has many drawbacks, especially in current requirements: portable devices games with a very low amount of computer power or last generation personal computer games. Current videogames are every time more and more similar to virtual reality applications: consumed resources, realism, complexity, techniques used or user interfaces. But, they still use old videogames simulation schemes, not virtual reality application schemes. This paper tries to show the current videogame kernels work better if they use virtual reality discrete approaches instead of continuous ones. A discrete approach defines an own private sampling period for every videogame object, adjusting the sampling to the object behavior. The objects events are executed ordered by time. There is no defined an objects priority. The render object works as any other videogame object. So, the SRR can be adjusted to the system load or characteristics.

References

1. Banks, J., Carson II, J.S., Nelson, B.B., Nicol, D.M., Discrete-Event System Simulation. Prentice Hall International Series in Industrial and Systems Engineering, 2001.
2. Buyasundial Page. http://www.buyasundial.com/.
3. Conitec Datasystems. http://www.conitec.net/.
4. Croteam. http://www.croteam.com/.
5. Crystal Space. http://crystal.sourceforge.net/drupal/.
6. Darken, R., Tonnesen, C., Passarella, K., The Bridge Between Developers and Virtual Environments: a Robust Virtual Environment System Architecture. Proceedings of SPIE 1995, No. 2409-30.
7. García, I., Mollá, R., Barella, A., GDESK: Game Discrete Event Simulation Kernel. WSCG'2004.
8. García, I., Mollá, R., Ramos, E., Fernández, M., D.E.S.K. Discrete Events Simulation Kernel. ECCOMAS'2000.
9. Genesis3D. http://www.genesis3d.com/.
10. Idsoftware Page. www.idsoftware.com/archives/doomarc.html.
11. Law, A.M., Kelton, W.D., Simulation Modeling and Analysis. McGraw-Hill Series in Industrial Engineering and Management Science, 1982.
12. Pausch, R., Burnette, T., Capehart, A.C., Conway, M., Cosgrove, D., DeLine, R., Durbin, J., Gossweiler, R., Koga, S., White, J., A Brief Architectural Overview of Alice, a Rapid Prototyping System for Virtual Environments. IEEE Computer Graphics and Applications, 1995.
13. Pritsker, A.A.B., The GASP IV Simulation Language. John Wiley and Sons, Inc., New York. 1974.
14. Quake Developers Page. www.gamers.org/dEngine/quake/.
15. Shaw, C., Liang, J., Green, M., Sun, Y., The Decoupled Simulation Model for Virtual Reality Systems. CHI'1992.
16. Tom's Hardware Guide. www6.tomshardware.com/graphic/02q1/020304/geforce4-09.html.
17. Watt, A., Policarpo, F., 3D Computer Games Technology: Real-Time Rendering and Software. Addison-Welsey, 2001.
18. Watt, A., Policarpo, F., 3D Computer Games. Addison-Wesley, Vol.2, 2003.

Speech Driven Facial Animation Using Chinese Mandarin Pronunciation Rules

Mingyu You, Jiajun Bu, Chun Chen, and Mingli Song

College of Computer Science, ZheJiang University, China

Abstract. This paper presents an integrated system aimed at synthesizing the facial animation from speech information. A network IFNET composed of context-dependent HMMs(Hidden Markov Model) representing Chinese sub-syllables is employed to obtain the corresponding Chinese initial and final sequence within the input speech. Instead of being based on some finite audio-visual database, IFNET is just built according to the Chinese mandarin pronunciation rules. Considering the large amount of computation, we embed Forward-Backward Search Algorithm in the course of searching in IFNET. After the initial and final sequence constructed, they are converted to the MPEG-4 high-level facial animation parameters to drive a 3D head model performing corresponding facial expressions. Experiment results show that our system works well in simulating the real mouth shapes, giving the speech information in many different situations speaking Chinese.

1 Introduction

Although the bandwidth of Internet grows rapidly, it still can not match the growth of Internet usage. On the other hand, users prefer to communicate with video and audio rather than with static information such as static image and text. Therefore, developing low bit-rate but high-resolution communication tools becomes more and more necessary. Compared with current frame-based videos, synthetic faces with synchronized voice appears to be a good method to reduce the bit-rate of communication. Because a remote computer can reconstruct the animation sequences with some key parameters. Besides, research shows that a synthetic talking face can help people understand the associated speech in noisy environments [1]. It also helps people react more positively in interactive services [2].

There are two ways for implementing the facial animation with original speech: via speech recognition or not via speech recognition. The latter method is to construct a direct mapping from speech acoustics (e.g. linear predictive codes) onto facial trajectories, either through control parameters of the facial animation itself [3], three-dimensional coordinates of facial points [4] or articulatory parameters [5]. Although this method has achieved reasonable results, we feel its performance can not match the request of simulating mouth movement accurately. Firstly, compared with the relationship between acoustic information and phoneme(or syllable), the connection between speech and facial animation

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 886–895, 2004.
© Springer-Verlag Berlin Heidelberg 2004

is looser and more complicated [6]. So it'll be harder to find appropriate acoustic features to distinguish different facial expression. Secondly, constructing a direct mapping from acoustic features onto facial animation will be a little eyeless and lose some information useful in synthesizing facial model. As we know that mouth is closed before pronouncing the Chinese initial 'b' in syllable 'baba'. But it's too hard for the direct mapping method to get this information which has no sign on the acoustic signal [7] and can only be gained after 'b' is recognized.

Synthesizing the facial animation via speech recognition proved to be simple and efficient [8] [9]. Yamamoto [8] and his fellows proposed a lip movement synthesis method based on extended HMMs. Though they obtained a good result on lip movement synthesis for Japanese words, a finite database was used in training. It leads to a result that the synthesis effect becomes awful if the test materials totally differ from the training database. I-Chen Lin [10] proposed a system using speech recognition engines to get utterance and corresponding time stamps in the speech data. He treated the mouth shape in a coarse way which just synthesizing a probable mouth movement, because he segmented the input speech into Chinese syllables. It is known that during pronouncing a Chinese syllable such as 'men', mouth changes from one shape to another instead of maintaining a static figure. So just segmenting speech into syllables is not advised.

In this paper, we propose to synthesize a 3D talking head with synchronized voice via speech recognition. Firstly, we train HMMs using acoustic information of Chinese initials and finals. Then, using the HMMs, we construct a network IFNET based on Chinese pronunciation rules instead of some finite audio-visual database. Finally, we form a Visubsyllable pool with mouth shape parameters from tracking results. During the course of synthesis, we find out the corresponding initial and final sequence from IFNET for input speech. Then, with mouth shape parameters for initials and finals, we synthesize facial animation with synchronized speech.

The remainder of the paper is organized as follows. First, we describe the main function of our system and how it works in section 2. And then we focus on the network IFNET in section 3. Section 4 shows the experiment result of our system. Finally, conclusion and acknowledgement are in section 5 and 6.

2 System Overview

Our system consists of two parts which are training and synthesis, respectively. As preprocessing, we train the HMMs with acoustic information of selected initials and finals segmented from training speech. Then, based on Chinese mandarin pronunciation rules, we construct the network IFNET with the HMMs. Besides preprocessing the acoustic information, we track the facial movement in video and find the appropriate Visubsyllables for the corresponding initials and finals. Through measuring the mouth shape parameters of the Visubsyllable, we calculate FAPs based on these parameters and form Visubsyllable pool.

In the synthesis part, we should also segment the acoustic signal and find out the corresponding initial and final sequence for input speech using network

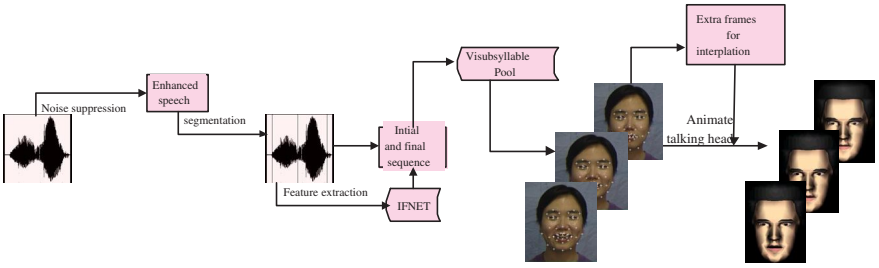


Fig. 1. System Overview

IFNET. With Visubsyllabel pool, we get the mouth shape parameters for key frames. After interpolating some extra frames for transition between key frames, we finally synthesize a 3D talking head with synchronized speech. The whole course is shown on figure 1.

3 Network-IFNET

3.1 Data Preparation

We record the video for training and testing on 40fps. In order to be speaker independent, 11 persons from different areas of China are invited to take part in our data preparation. Six of them are male and five are female. Each subject is asked to read a paragraph designed for training and a paragraph for testing.

3.2 Signal Processing and HMM Training

For English speech recognition, people traditionally segment the acoustic sequence into syllables [11] or phonemes [12]. A syllable is the next bigger unit of speech after a phoneme in English, but it's different in Chinese which has a initial or final between a syllable and a phoneme. As we mentioned in the introduction section, dividing speech into syllables can not satisfy the request of synthesizing the mouth movement accurately. On the other hand, phoneme sequence is much longer than initial and final sequence, it'll need more complicated computation to construct appropriate phoneme sequence for output. Based on the reasons mentioned above, we choose Chinese initials and finals as the recognition units.

All input speech signals are pre-emphasized with a high-pass filter $1\text{-}0.98z^{-1}$ and then digitized into 8-bit data format at a rate of 22 kHZ. We segment the input speech into initial and final sequence based on the short-time energy (1) and the zero-crossing ratio of the signal. Then a 25ms Hamming window is applied to the signal every 10ms interval followed by the 12-order MFCC cepstrum analysis. MFCC(Mel-Frequency Cepstral Coefficients)[13] we employed here are known to be useful for speech recognition and robust to variations among speakers and recording conditions.

$$E = \log \sum_{n=1}^{N} s_n^2 \tag{1}$$

There are two acoustic models used widely, HMM(Hidden Markov Model) [14] and NN(Neural Network) [15]. Here, we choose HMM for its good performance in modelling the evolution of audio-visual speech parameters [16]. We use 5 states left-to-right HMM model in our system. It is known that in continuous speech, pronunciation of a syllable is influenced by both the proceeding and the following. So we employ the context-dependent HMM in order to exactly recognize the input speech, for example, we use different HMM for 'b' in 'bi' and 'bo'. In order to achieve good performance for continuous speech, we form a HMM sequence based on the transcription of the training sentence when training HMMs. For example, if the transcription of the sentence is 'ba he ma', we connect the HMM for 'b', HMM for 'a' and so on. Then we train HMMs using the Baum-Welch algorithm [17] by mapping the MFCC coefficients to the corresponding initial or final.

3.3 Visubsyllable Pool Construction

Traditionally the speech animation of 3D synthetic faces involves extraction of visemes which are visual counterpart of phonemes. But in this paper, we propose to use Chinese initials and finals as the recognition units, so we take a novel approach for speech animation - using Visubsyllables, the visual counterpart of subsyllables including Chinese initials and finals. When we segment the input speech into initial and final sequence in the signal processing part, we get a line of time stamps representing the start and end of every subsyllable. Here we cut the video into segments using the same time stamps. Among frames in the segment for a initial or final, we should select one as the Visubsyllable to represent the mouth shape when pronouncing the initial or final. For Chinese initial, mouth shapes up before phonation, so we select the frame just before the short-time energy of signal grows(marked in the left of figure 2). We pick the frame for Chinese final when the short-time energy of signal reaches peak(marked in the right of figure 2), because mouth shape is stablest at that moment. At the last step, we measure the feature points of Visubsyllables such as the points

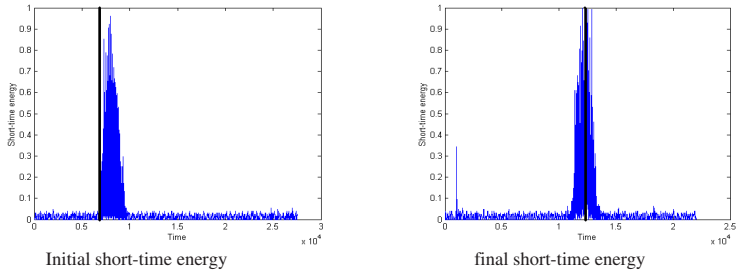


Fig. 2. Initial and final waveform for visubllyable extraction(the black thick line marked on the waveform represents the position we select a frame for the corresponding subsyllable)

around mouth by hand ,compute the average values from different persons and calculate the corresponding FAPs to construct the Visubsyllable pool. In our system, we select 20 FAPs which is mostly focus on the mouth from MPEG-4 facial animation parameters.

3.4 IFNET

With the HMMs representing Chinese initials and finals, we can simply segment the input speech and recognize every segment separately. But this method will result in a awful performance, for example, any noise in a segment will bring to a recognition mistake. Besides, recognition separately will output some initial and final conjunction that doesn't exit at all. So we turn to recognize the segment sequence together. In this way, we should check all possible HMM conjunctions and find one of them most suitable to recognize what the speaker said. To form the HMM conjunctions, we must know which HMM can follow the current one. We can not try all the HMMs representing initials and finals because it'll make the computation too complicated. Many people employ a network instead. In the network, every node represents a HMM and linked by nodes representing the HMMs which could follow the current one. How to construct the network has been a pop research area. Some researchers just simply focus on particular applications and construct the network by the rules of word connection in that situation. Because this method is unable to construct a general system, many researchers turn to form the network using large audio database. They study the database, summarize some rules for syllable conjunction and construct network based on these rules. This method appears to be a good way for building a general system, but it's arduous to summarize the word connection rules and the result is dependent on the researcher's experience. Moreover, the rules summarized is directly influenced by the scale and the scope of the database.

Different from the approaches used before, we prefer to build a network based on the Chinese pronunciation rules. Chinese is a language composed of characters. And in order to standardize Chinese pronunciation around the country, experts put the Chinese syllables forward. Chinese syllable can divide into Chinese initial and final. It is know that the conjunction between the initial and final should obey some rules. There are totally 21 initials and 39 finals in Chinese. We summarize the rule for which final can follow the appointed initial into a table. A part of our table is shown in table 1.

With the rules summarized above, we can construct the IFNET as follows. Firstly, we put all the initials and zero-initial in the first layer. Here we use the phrase zero-initial to refer to the initial in syllable without initial. In the second layer, we allocate the finals and connect the legal finals to every initial in the first layer. Legal finals represent the finals that can follow certain initial according to the conjunction rules. After passing through the two layers, we have two choices, still come back to the first layer as if it was the third layer or just simply exit IFNET. We offer a visual description of IFNET in Figure 3.

After acquiring the segment sequence, we use IFNET to recognize the initial or final for every segment. Firstly, we use the Viterbi algorithm to compute the

Table 1. Part of the table for conjunction of Chinese initials and finals. (The ✓ in the grid means the final in the column it stands can follow the initial in the same row with it. And the empty grid shows the final that column can not follow the initial stands the same row as the grid.)

Initials	Finals																					
	a	o	e	ai	...	ia	ie	iao	iu	...	uai	ui	uan	un	uang	...	ü	üe	üan	ün	iong
b	✓	✓		✓	...		✓	✓		...						...					
p	✓	✓		✓	...		✓	✓		...						...					
m	✓	✓	✓	✓	...		✓	✓	✓	...						...					
f	✓	✓			...					...						...					
d	✓		✓	✓	...		✓	✓	✓	...		✓	✓	✓		...					
⋮												⋮									
j					...	✓	✓	✓	✓	...		✓	✓	✓		...	✓	✓	✓	✓	✓
q					...	✓	✓	✓	✓	...		✓	✓	✓		...	✓	✓	✓	✓	✓
x					...	✓	✓	✓	✓	...		✓	✓	✓		...	✓	✓	✓	✓	✓
⋮												⋮									

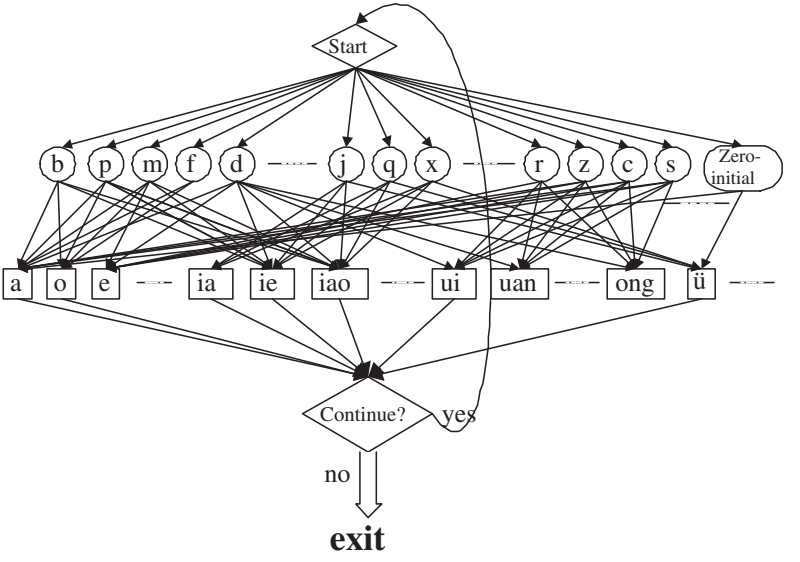


Fig. 3. part of the network-IFNET(Initials and finals are in layer one and two separately, the suspension points between the radials adhere to the zero-initial means that all the finals can follow the zero-initial.)

probability for every HMM in the first layer of IFNET to output the observation sequence of the first segment. Then, we continue to compute the probability for HMMs in the second layer with the observation sequence of the second segment as their output. Multiplying the probability of two HMMs with links in IFNET,

we can get the probability for all the correct initial and final conjunctions. If some product of the multiplication under the pruning threshold we set, we just discard these conjunctions. With the remainder, we should judge whether to exit IFNET or come back to the position we start in IFNET based on the left segment sequence. It's a rotative process and at the exit of IFNET, we select the HMM sequence (representing the initial and final sequence)with the largest probability product.

Considering the large amount of computation in searching IFNET, we employ Beam Search to prune HMM conjunctions far from being the recognition result. Pruning threshold is very important in Beam Search, but selecting its value is a hard work. If the value is too large, we may discard the best sequence only because its imperfect performance at the beginning. On the contrary, if we set the value too small, it'll lead us to do a lot of useless job. So we adopt the Forward-Backward Search Algorithm [19]in search IFNET. The algorithm takes place in two phases. The first phase performs a fast time-synchronized search of the utterance in the forward direction. The backward pass performs a more expensive search, processing the utterance in the reverse direction and using information gathered by the forward pass. Whether the path will be discarded depends on its performance at the whole process instead of at the beginning.

3.5 3D Talking Head Synthesis

With the recognized initial and final sequence, we can find the corresponding Visubsyllable sequence from the Visubsyllable pool as the key frames for facial animation. It's obvious that simply animating 3D face model with these key frames will get an awful result. So we should interpolate some extra frames between two key frames. Here we calculate parameters of the mouth (such as upper lip position, lip width, jaw rotation, tongue, etc.) of a given frame to be interpolated by integrating weighted contributions of frames stand nearby [3]. Using all these frames and the time stamps for every frame, we can finally synthesize facial animation with synchronized speech vividly(Figure 4).

Fig. 4. An example of the synthesized face sequence. The 3D synthesis face is pronouncing Chinese phrase "wo men".

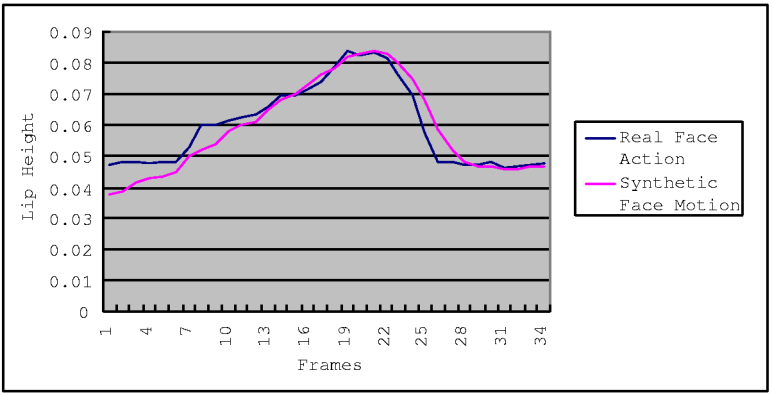


Fig. 5. Comparison for the lip height of real face action and synthetic face motion when pronouncing Chinese syllable "wo".

4 Experiment Results

In order to judge the subjective quality of the animations, we employ two methods. One of them is evaluations participated by native viewers. we design a set of blind trials in which native observers tried to distinguish synthesized from real facial motion. We take 1200 frames of tracked video designed for testing, and synthesize new facial motion from the audio. Then we generate animations from both the synthesized motion and the "ground-truth" tracked motion. After breaking each animation into three segments, we present all segments in random order to naive observers. The subjects are asked to select the "more natural" animations. There are 12 persons taking part in the evaluation. Three observers consistently prefer the synthesized animation; four consistently prefer the ground-truth animation; and other persons' judgements shift between the two animations with 38% prefer the synthesized animation. The experiment results indicate that while synthesized facial action can be distinguished from the true action(real facial action is more varied), they are almost equally plausible to naive viewers.

Besides, we also design an experiment to quantificationally evaluate the performance of our system. Considering the lip height can mostly represent the mouth shape, we choose to compare the lip heights of the synthesized faces with the original ones. Figure 5 indicates that the synthesized result match the real action accurately.

5 Conclusion and Future Work

In this paper, we present a way to synthesize a 3D talking head with the input speech. By implementing facial animation via speech recognition, we find a simple but effective way to synthesize a talking head. By choosing Chinese initials

and finals as the recognition units, we can obtain realistic and accurate synthesized animations. By employing a network-IFNET, our system can be applied to any condition speaking Chinese.

Although our system based on IFNET reports good performance and the speed for searching IFNET is acceptable, it still need to be improved to match the request of real-time synthesis. We'll continue to refine IFNET by more statistic results of Chinese pronunciation. Besides, some novel and effective algorithm for searching IFNET is advised.

Acknowledgement. This work is partly supported by NSFC grants 60203013 and HP laboratory of ZheJiang University.

References

1. D.W.Massaro: "Speech Perception by Ear and Eye: A Paradigm for Psychological Inquiry". Hillsdale, NJ: Lawrence Erlbaum, 1987.
2. I. Pandzic, J. Ostermann and D. Millen: "User evaluation: Synthetic talking faces for interactive services". Visual Comput., vol. 15, no. 7/8, pp. 330–340, 1999.
3. M. Cohen and D. Massaro: "Modeling coarticulation in synthetic visual speech." Models and Techniques in Computer Animation, pp. 141–155. Springer Verlag, Tokyo, 1993.
4. M. Brand: "Voice Puppetry." Proceedings of SIGGRAPH'99, pp. 21–28, 1999.
5. F. Lavagetto: "Converting speech into lip movements: A multimedia telephone for hard of hearing people." IEEE Transactions on Rehabilitation Engineering 3(1), pp. 90–102, 1995.
6. Yi-qiang CHEN: "Data Mining and Speech Driven Face Animation." Journal of System Simulation, Vol.14 No.4, pp. 496–500, 2002.
7. Zhiming Wang and Lianhong Cai: "Study of the Relationship Between Chinese Speech and Mouth Shape." Journal of System Simulation, Vol.14 No.4, 2002.
8. E.Yamamoto, S.Nakamura and K.Shikano: "Lip movement synthesis from speech based on hidden Markov models." Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 154–159, 1998.
9. K.Waters and T.Levergood: "DECface:A system for synthetic face applications." Multimedia Tools and Applications, vol.1, No.4, pp. 349–366, 1995.
10. I-Chen Lin, Chen-Sheng Hung, Tzong-Jer Yang and Ming Ouhyoung: "A Speech Driven Talking Head System Based on a Single Face Image." Proc. Pacific Conference on Computer Graphics and Applications '99 (PG'99), pp. 43–49, 1999.
11. Sumedha Kshirsagar and Nadia Magnenat-Thalmann: "Visyllable Based Speech Animation." EUROGRAPHICS 2003, Volume 22(2003), Number 3,2003.
12. Christoph Bregler, Michele Covell and Malcolm Slaney: "Video Rewrite: Driving visual speech with audio." Proceedings of the 1997 Conference on Computer Graphics, SIGGRAPH, pp. 353–360, 1997.
13. S.Davis and P.Mermelstein: "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." Proceedings of IEEE International Conference on Acoustics, Speech and Signal, pp.357–366, 1980.
14. Taisuke Itoh, Kazuya Takeda and Fumitada Itakura: "Acoustic analysis and recognition of whispered speech." Proceedings of IEEE International Conference on Acoustics, Speech and Signal, pp. I/389–I/392, 2002.

15. K.Kasper, H.Reininger, D.Wolf and H.Wuest: "Fully recurrent neural network for recognition of noisy telephone speech." Proceedings of IEEE International Conference on Acoustics, Speech and Signal, pp. 3331–3334, 1995.
16. R. Rao and T. Chen: "Using HMM's for Audio-to-Visual Conversion." IEEE Workshop on Multimedia Signal Processing, 1997.
17. L.R.Rabiner: "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE, Volume:77, Issue:2, pp.257–286, 1989.
18. S.Boll: "Suppression of acoustic noise in speech using spectral subtraction." Proceedings of Acoustics, Speech, and Signal Processing, Volume:27, Issue:2, pp.113–120, 1979.
19. S.Austin, R.Schwartz, P.Placeway: "The forward-backward search algorithm." Proceedings of Acoustics, Speech, and Signal Processing, vol.1, pp.697–700, 1991.

Autonomic Protection System Using Adaptive Security Policy*

Sihn-hye Park[1], Wonil Kim[2**], and Dong-kyoo Kim[3]

[1] Graduate School of Information and Communication, Ajou University, Suwon, 443-749, Republic of Korea
sinne@ajou.ac.kr
[2] College of Electronics and Information Engineering, Sejong University, Seoul, 143-747, Republic of Korea
wikim@sejong.ac.kr
[3] College of Information Technology, Ajou University, Suwon, 443-749, Republic of Korea
dkkim@ajou.ac.kr

Abstract. There are various techniques to safeguard computer systems and networks against illegal actions. Secure OS based on Role Based Access Control (RBAC) is one of the systems that reflect these techniques. The RBAC system controls access to system resources based on roles. Recently, many systems employ more fine-grained access control on system resources to enhance system security. However, this approach in access control level may cause unexpected problems, since most systems acquire system resources through system call that is hooked on kernel. In this paper, we propose a novel approach to Intrusion Detection System (IDS). The proposed Autonomic Protection System (APS) supports fine-grained intrusion detection. It resides above Secure OS based on RBAC that provides general-grained access control. The system detects intrusions using security policy based on RBAC model. The system performs double checking for intrusions using positive and negative intrusion detection policy. Additionally, as one of active responses, the system supports the self-adaptation of security policy depending on various computing environments. Therefore, the system can detect intrusions more accurately and respond to the attacks actively and appropriately.

1 Introduction

These days, there are various security mechanisms to protect computer systems and networks against internal and external attacks. Secure OS (Secure Operating System) based on RBAC (Role Based Access Control) that controls access to system resources based on roles is one of the systems that reflect these techniques. To enhance system security, the RBAC systems use more fine-grained access control on system resources. Actually, these approaches in access control level may cause unexpected problems of system performance and safety such as high system overload, halt and

* This study was supported by the Brain Korea 21 Project in 2004.
** Author for Correspondence, +82-2-3408-3795

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 896–905, 2004.
© Springer-Verlag Berlin Heidelberg 2004

down. This may be caused by the fact that most Secure OS controls access to system resources using system call that is hooked on kernel level. The problems for these approaches in access control level can be solved in intrusion detection level. Therefore, we employ two-level approach that fine-grained approach on intrusion detection level and general-grained approach on access control level. In this paper, we present the approach to security management considering system overload, performance, and safety.

The remainder of this paper is organized as follows. Section 2 presents the background researches of the proposed system such as RBAC system, Intrusion Detection System, and Autonomic Computing. Section 3 proposes the novel approach for Intrusion Detection System (IDS). Section 4 simulates the proposed system, and section 5 concludes.

2 Background Researches

2.1 RBAC System

RBAC (Role-Based Access Control) system controls access to system resources based on roles. As the system enforces security policy based on roles, the modeling of complex access control policies is possible and the cost of administration and administrative errors can be reduced [1][2]. RBAC determines if users can be granted a specific type of access to a resource. In an RBAC system, users are granted or denied access to a device or service based on the role that users have as part of an organization [3].

The RBAC model taxonomy consists of four models such as core RBAC, hierarchical RBAC, static constrained RBAC, and dynamic constrained RBAC [4]. There are various concepts such as role hierarchies, separation of duty, constraints, and so on. In this paper, we discuss only core RBAC model and its basic concept. Core RBAC refers to the basic set of features that are included in all RBAC systems. As shown in Fig. 1, core RBAC consists of five administrative elements such as users, roles, permissions, operations, and objects. Permissions are associated with roles, and users are assigned as members of roles. By this way, users can acquire the roles' permission. This approach provides great flexibility and granularity of security management [4].

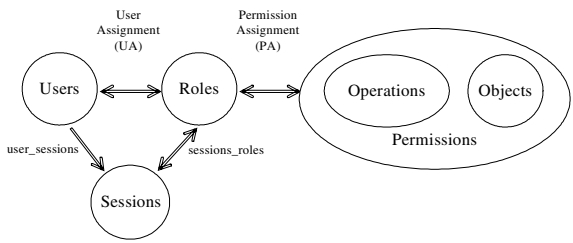


Fig. 1. Core RBAC model

2.2 Intrusion Detection System

Intrusions are defined as any unauthorized attempts to access, manipulate, modify, or destroy information, or to render a system unreliable or unusable [5]. Many researches for intrusion detection were done in order to improve intrusion detection techniques since Jim Anderson initiated in the early 1980s [5].

Intrusion detection techniques are generally categorized by two approaches: misuse detection and anomaly detection. Misuse detection technique defines specific patterns on a system and detects intrusions by the patterns [6]. The patterns indicate previous behaviors and actions that were deemed intrusive or malicious. This method has difficulties to detect novel attacks and may cause the high rate of missed attack (false negatives). However, it is easy to determine whether a specific action is intrusive or not. Therefore, today, the majority of intrusion detection systems adopt misuse detection systems that identify attacks based on attack signatures. On the other hand, anomaly detection technique defines the baseline of the normal behavior on a system and detects intrusions using deviations from this baseline. The method can detect new attacks against systems by compare current activities against statistical models of past behavior. However, the system may generate too many false alarms (false positives) by narrowly defining the baseline of normal behaviors [5]. As another method of intrusion detection, policy-driven intrusion detection is proposed by Suresh N. Chari et al. recently [7]. Policy-driven technique defines the boundary between the good and bad as a set of rules. Policy-driven intrusion detection system creates an infrastructure for defining and enforcing very fine-grained process capabilities in kernel. Process capabilities are referred as a set of rules, for controlling access to system resources. This method offers a number of advantages to solve the above problems of the traditional detection methods. However, there are disadvantages such as version migration and detection problem of memory attacks since the rules are defined only on access to system resources [7].

2.3 Autonomic Computing

Proper managements of computer systems and networks are very difficult due to their increasing complexity. There has been exponential growth in the number and variety of computer systems and distributed networks [8]. Moreover, integrating several heterogeneous environments and extending company boundaries into the Internet make computer systems and networks more complex [9]; as a result of it, system administrators have to manage massive and complicate tasks of configuring, optimizing, and maintaining [10]. Autonomic computing is an approach to self-managed computing systems which human does not intervene in their actions [11]. It can minimize the burden of administrators and implement computing systems running by themselves. It can also adjust to varying circumstances and handle their resources effectively [11].

Autonomic computing systems automatically configure, optimize, heal, and protect by themselves [10]. The self-management liberates system administrator from the maintenance of complicate systems. Additionally, it provides high performance and efficiency as it searches for optimum values continually [10]. Autonomic computing systems maintain their operations and adjust to varying internal or external conditions [10]. The systems also continuously monitor the status of systems in run time to

capture the necessity of system upgrade, software or hardware failure, external attack, and other problems [8][10]. These self-managements are considered as self-configuring, self-healing, self-optimizing, and self-protecting.

3 Proposed System: Autonomic Protection System

This section presents the architecture of the proposed Autonomic Protection System. Autonomic Protection System (APS) is defined as the system which does not only detect intrusions and respond to the intrusions, but also adapt automatically intrusion detection policy depending on various computing environments [12].

This system operates as high-level application above Secure OS based on RBAC. The Secure OS mediates accesses to the kernel's internal objects: tasks, inodes, open files, etc [2][13]. By this reason, the attempts to enforce more fine-grained policy may cause not only the problems of system performance, but also the critical accidents such as memory overload, system halt and system down. Contrary to the RBAC system, the proposed APS has minimal impact on system performance because it just monitors accesses to the kernel's objects, and analyze them. Therefore, the proposed system and the RBAC system interact with each other to protect a computer system. In other words, APS supports more secure and fine-grained security activities as the system works above RBAC system that interacts with it.

The system detects intrusions using positive/negative intrusion detection policies. If an intrusion is detected, the system actively reacts with proper response. For example, the system can stop illegal session or send a request to RBAC system to modify security policy on critical security problems. Afterwards, the results of intrusion detection are verified by neural network. According to the verified results, the system modifies appropriately intrusion detection policy. By this way, the security policy in APS is adapted automatically. The goal of APS is to improve the security of the system gradually. As the system evolves, it will detect intrusions more accurately and adaptively.

The system consists of Intrusion Detection Module (IDM), Intrusion Response Module (IRM), Intrusion Verification Module, Intrusion Detection Policy (IDP), and Intrusion Response Policy (IRP) as shown in Fig. 2. More details are presented the following subsections.

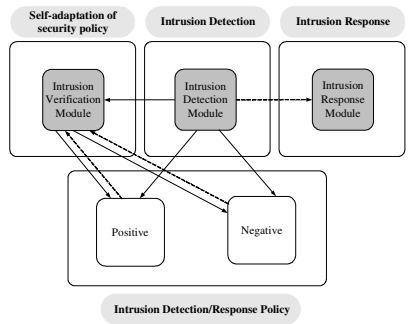


Fig. 2. Architecture of Autonomic Protection System

3.1 Intrusion Detection Policy

The security policy in the proposed system refers to intrusion detection and response policy. The security policy in the system is based on RBAC model in the sense that the system has the concepts of subjects and objects as RBAC model has. It basically includes core RBAC static element such as users, roles, and permissions. Permissions consist of objects and operations. Therefore, it is constructed as the security policy of RBAC system except the concept of security enforcement.

Intrusion detection policy is composed of two policies, positive intrusion detection policy and negative intrusion detection policy. Positive intrusion detection policy describes unallowable roles meaning intrusion in a system. On the other hand, negative intrusion detection policy describes allowable roles meaning non-intrusion. The system detects intrusion by the two policies. When an action occurs, the system observes negative intrusion detection policy, after which it examines positive intrusion detection policy. Especially, positive intrusion detection policy is valuable because it is not only used to reduce false negatives by multiple checking of intrusions from different aspects, but also used to generate trained verification data set. This data set is important to verify intrusion results in the proposed system.

The binary codes representing two policies are trained by neural network. After training, verification data set is generated. When unexpected exception occurs or new program is installed, the results of intrusion detection are needed to be examined. Afterwards, if necessary, intrusion detection policies are modified using verification information. By this way, the security policy in the system is adapted by itself.

3.2 Intrusion Detection Module

The proposed system collects data from network traffic, system logs, and hardware reports. After collecting data, it analyzes the information and generates action streams according to the format of intrusion detection policy. The system detects intrusions using the action streams based on two policies; positive/negative intrusion detection policies. Initially, an action stream is compared with negative policy, then, it is compared with positive policy.

The following Table 1 shows the decision whether an intrusion occurs or not. In the first row, an action stream is matched with two policies. It indicates that positive policy and negative policy are conflicted with each other. In this case, the action stream is decided to intrusion and it is sent to Intrusion Verification Module (IVM) since the corresponding policy is needed to be verified. In second and third row, action streams are matched with positive policy or negative policy respectively. In this case, each action stream is decided to intrusion or non-intrusion respectively. In fourth row, an actions stream is not matched with two policies. It indicates the case that a user performs unexpected actions or new application is installed or configured. In case of that, the system sends the information to Intrusion Verification Module (IVM) in the same case of the first row. This will be discussed in detail in section 3.4. With this double checking that uses positive and negative policies, the system can reduce false positives and false negatives.

Table 1. Intrusion Detection using Positive/Negative Intrusion Detection Policies

Positive policy	Negative policy	Intrusion (I) / non-intrusion (N)	Description
Y	Y	I	Positive policy and negative policy are conflicted with each other.
Y	N	I	-
N	Y	N	-
N	N	I	New program is installed or a user performs unexpected actions.

3.3 Intrusion Response Module

In order to have secure system, it is important to respond to attacks appropriately as well as to detect intrusions accurately. This means that both intrusion detection system and intrusion response systems have to work in parallel [14][15][16]. It also should be emphasized how to respond in accordance with the characteristics of the detected attack. When a suspicious activity is detected, various responses according to the degree of security-level should be considered.

The proposed system responds to attacks according to intrusion response policy in Intrusion Response Module (IRM). Basically, when an intrusion occurs, the predefined responses associated with positive intrusion detection policy are performed.

When new positive intrusion detection policy is created, corresponding responses associated with the policy are predefined. To decide the responses, IRM employs neural network. The IRM consists of input, hidden, and output layers. Input layer consists of the input nodes that indicate the binary bits of an action stream, the results of intrusion detection. Output layer has the decision nodes that indicates responses adopted in the system. The output nodes are represented between '1' and '0'. If the value of an output node is closed to intrusion, it is closed to '1', otherwise, it is closed to '0'. This threshold values will be determined according to computing environments. IRM learns intrusion patterns using predefined response policies associated with positive intrusion detection policy. It is used to decide what responses are performed for an intrusion.

3.4 Intrusion Verification Module

One of the difficulties recognized in Intrusion Detection System is determining the proper level of security management. To solve this problem, the security policy of IDS is required to be well-defined in their environments. Security policy created by software developers or system administrators manually can not be absolutely perfect. Therefore, to reduce false rates, the security policy of IDS is necessary to reconfigured according to the status of computing environments.

Especially, when exceptions in intrusion detection occur, intrusion detection policy is needed to be examined and modified. Exceptions are the followings: (1) unexpected action occurs, (2) other programs are installed, or (3) the configuration of a system is changed. When above exceptions occur, the result of intrusion detection is sent to intrusion verification module (IVM). Intrusion verification module decides whether the given result is intrusion or not. If the decision is intrusion, the positive policy is

needed to be modified, otherwise, the negative policy is needed to be changed. Additionally, intrusion detection policy is also needed to be examined and modified, when positive and negative intrusion detection policies are conflicted with each other due to errors by administrators.

The intrusion verification module (IVM) employs neural network as in IRM. The Input layer of IVM consists of the binary bits of the results of intrusion detection. Output layer has one decision nodes. IVM learns intrusion patterns using predefined positive/negative intrusion detection policies. It is used to decide whether intrusion occurs or not. After verification through this way, intrusion detection policy is modified appropriately.

4 Simulation

4.1 Intrusion Detection Policy

The following Table 2 and Table 3 show the sample of positive intrusion detection policy and negative intrusion detection policy for simulation respectively. The example of the fourth row of Table 2 and the example of the first row of Table 3 are illustrated in the first and second row of Table 4 respectively.

Table 2. Example: Positive Intrusion Detection Policy

Policy	Roles	Users	Permissions		Response
			Object	Operations	
role_sys_adm	0001	1001	0001	1110	000001
role_db_adm	0010	1010	0010	1101	000101
role_dept_1	0011	1011	0011	1100	001001
role_gen_usr	0100	1100	0100	1011	001101
role_unauth_usr	0101	1101	0101	1010	010001

Table 3. Negative Intrusion Detection and Response Policy

Policy	Roles	Users	Permissions	
			Object	Operations
role_sys_adm	0001	1001	0001	0001
role_db_adm	0010	1010	0010	0010
role_dept_1	0011	1011	0011	0011
role_gen_usr	0100	1100	0100	0100
role_unauth_usr	0101	1101	0101	0101

4.2 Intrusion Detection and Response

The following Table 4 shows the results of intrusion detection using two policies. In intrusion detection module, each action stream is generated according to the method of intrusion detection (described in section 3.2) whether it is intrusion or not. That means, if two bits of the action stream are “11”, it is intrusion. This case indicates that two policies conflict with each other. In this case, action streams are needed to be verified. If the two bits are “10”, it is not intrusion. If the two bits are “01”, it is

Table 4. The results of intrusion detection

Roles	Users	Objects	Opera-tions	Positive Policy	Negative Policy	Intrusion/Non-intrusion
0100	1100	0100	1011	1	0	I
0001	1001	0001	0001	0	1	N
0011	1011	0011	1101	0	0	I
0000	1010	0000	0001	0	0	I

Table 5. Intrusion response actions

Response	Description
000001	Generate an Alarm
000010	Lock User Account
000100	Suspend User Jobs
001000	Terminate User Session
010000	Request to Modify Security Policy to RBAC system
100000	Verify and Modify Intrusion Detection Policy

Table 6. The verification results of action stream

Roles	Users	Objects	Operations	Positive Policy	Negative Policy	Verification
0011	1011	0011	1101	0	0	0.444980
0000	1010	0000	0001	0	0	0.991248

Table 7. The modification of positive intrusion detection policy

Roles	Users	Permissions		Response
		Object	Operations	
0001	1001	0001	1110	000001
0010	1010	0010	1101	000101
0011	1011	0011	1100	001001
0100	1100	0100	1011	001101
0101	1101	0101	1010	010001
0011	1011	0110	0001	000001

intrusion. In this case, appropriate responses are performed. If the two bits are “00”, it is intrusion. Especially, the two bits “00” of the third and the fourth row in Table 5 indicate special cases. In this case, action streams are also needed to be verified. Therefore, the action streams are sent to intrusion verification module.

When intrusions occur, the system performs response actions corresponding to intrusion response policy associated with positive intrusion detection policy. In this simulation, we used six intrusion response actions as shown in Table 5.

4.3 Intrusion Verification

In intrusion verification module (IVM), intrusion detection policies are verified and modified. By predefined threshold value, if an action stream indicates intrusion, then the action stream is registered in positive intrusion detection policy, otherwise, it is registered in negative intrusion detection policy. In the simulation, the threshold value to verify action stream is 0.7 by experiment. The following Table 6 shows the

verification results of intrusion detection. In the Table 6, first row indicates non-intrusion and second row indicates intrusion. Therefore, action stream "0011101100111101" is defined as non-intrusion and registered in negative intrusion detection policy. On the other hand, action stream "0000101000000001" is defined as intrusion and registered in positive intrusion detection policy. In that case, since new program is installed or system configurations are reconfigured, the followings are performed. Initially, appropriate role is decided depending on user information. That is, user ID of "1011" is matched to role ID of "0011". Next, new object is created. That is, new object ID of "0110" that does not assigned to any objects is created. After new object ID is determined, the object and operation is assigned to role. Afterward, appropriate response is decided by using trained data in IRM. The response policy is associated with positive detection policy. Finally, new stream "00111011011000010000001" of intrusion detection policy is generated, and the policy is registered in positive detection policy as shown in Table 7.

As these processes are repeated, the system has more secure and optimized policy in autonomous computing environment. Therefore, the system detects intrusions accurately and makes responses adaptively.

5 Conclusions and Future Work

In this paper, we proposed Autonomic Protection System (APS) that does not only detect intrusions and respond to the attacks by itself, but also adapt automatically intrusion detection policy depending on various environments. The method of double checking using positive and negative intrusion detection policy is used to reduce the false detection rate. Additionally, the system has the intrusion detection policy based on RBAC Model and complementarily interacts with RBAC system to safeguard computer systems considering security, performance, and safety.

We simulated the proposed APS and described the whole process of APS. These processes enhance system security adaptively and gradually. Simulations in the proposed system shows that propose APS can detect intrusions more accurately and respond to intrusions appropriately. Moreover, security policy of the system can be adapted depending on various computing environments.

In the future, the system will extend to security policy reflected various concepts on RBAC model such as role hierarchies, constraints, and separation duty. Additionally, the system will be implemented gradually and verified focusing on reasonable interaction with RBAC system.

References

1. Along Lin: Integrating Policy-Driven Role Based Access Control with the Common Data Security Architecture. HP Labs Technical Reports, HPL-1999-59, 990430, External (1999)
2. C. Wright, C. Cowan, J. Morris, S.Smalley, G.Kroah-Hartman: Linux Security Module Framework. http://www.kroah.com/linux/talks/ols_2002_lsm_paper/lsm.pdf (2002)
3. Overview: Cisco Administrative Policy Engine. http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cape/admin_gd/ovrvw_ad.htm

4. David F.Ferraiolo, D.Richard Kuhn, Ramaswamy Chandramouli: Role-Based Access Control. ARTECH HOUSE, INC (2003)
5. Anup K. Ghosh, Aaron Schwartzbard: A study in using neural networks for anomaly and misuse detection. In Proceeding of the 8th USENIX Security Symposium, Washington, D.C., USA, August 23–26 (1999)
6. Zheng Zhang, Jun Li, C.N.Manikopoulos, Jay Jorgenson, Jose Ucles: HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification. In Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY (2001)
7. Suresh N. Chari, Pau-Chen Cheng: BlueBox: A Policy-driven, Host-Based Intrusion Detection System. ACM Transactions on Information and System Security (TISSEC), Vol. 6, No. 2 (2003) 173–200
8. A.G.Ganek, T.A.Corbi: The dawning of the autonomic computing era. IBM SYSTEMS JOURNAL, Vol 42, No 1 (2003)
9. Evaristus Mainsah: Autonomic computing: the next era of computing. Electronics & Communication Engineering Journal (2002)
10. Jeffrey O.Kephart, David M.Chess: The Vision of Autonomic Computing. the IEEE Computer Society (2003)
11. Autonomic Computing Overview Questions & Answers. http://www.research.ibm.com/autonomic/overview/faqs.html
12. Sihn-hye Park, Wonil Kim, Dong-kyoo Kim: Agent-Based Protection System in Autonomic Computing Environment. In Proceedings of PRIMA2003 (2003) 117–128
13. James Stanger, Patrick T.Lane: Hack Proofing Linux: A Guide to Open Source Security. Syngress Publishing, Inc. (2001)
14. Noria Foukia, Salima Hassas, Serge Fenet. Jarle Hulaas: An Intrusion Response An Intrusion Response Scheme: Tracking the Source Using the Stigmergy Paradigm. In proceedings of SECURITY OF MOBILE MULTIAGENT SYSTEMS WS (SEMAS-2002) (2002)
15. Curtis A. Carver Jr.: Intrusion Response Systems: A Survey. Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA (2001)
16. W. Jansen, P. Mell, T. Karygiannis, and D. Marks: Mobile agents in intrusion detection and response. In 12th Annual Canadian Information Technology Security Symposium, Ottowa, Canada (2000)

A Novel Method to Support User's Consent in Usage Control for Stable Trust in E-business*

Gunhee Lee[1], Wonil Kim[2**], and Dong-kyoo Kim[3]

[1] Graduate School of Information Communication, Ajou University, Suwon, Korea
icezzoco@ajou.ac.kr
[2] College of Electronics and Information Engineering, Sejong University, Seoul, Korea
wikim@sejong.ajou.ac.kr
[3] College of Information Technologies, Ajou University, Suwon, Korea
dkkim@ajou.ac.kr

Abstract. In the recent Web environment, the key issue is not who can use the resource, but how to control the usage of resources. Usage control (UCON) is a consolidate solution for this problem. UCON can be applied to solve existing authorization problems and support finer-grained control than any other methods. However, there are many situations in e-business and information systems in which the user consent is needed. For example, a contents provider should obtain the consent of the originator in order to modify digital contents. With this trend, the consent is suggested as a new issue that can establish trust in e-business. Unfortunately, the current UCON model does not support user consent. In this paper, we propose a novel method to control user consent in UCON model. We modify the authorization model of the UCON model to support consent handling and discuss the administration of proposed model. The proposed method extends the coverage of UCON model in security area. It also provides a solution for trust relationships on e-Business and individual privacy protection.

1 Introduction

The technological innovations in information processing have changed the life style of people and created new e-business model. In companies, schools, and public offices, people are hooked on computers to connect to the Web. Not only hypertext type contents but also digital contents, such as movie, music, and e-book are provided via Internet. As the Web environment becomes more complicated, we begin to encounter more vulnerable cases, such as distribution of illegally copied contents that should be handled at the various points of view. In order to satisfy all of these cases, various authorization and authentication methods such as access control, trust management, and digital rights management (DRM) are proposed. While access control makes authorization decision based on the identity of the resource requester, trust management does it based

* This study was supported by the Brain Korea 21 Project in 2003.
** Author for corresponding, +82-2-3408-3795

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 906–914, 2004.
© Springer-Verlag Berlin Heidelberg 2004

on the trust relationships [1]. Therefore, in trust management, even though the server does not know the identity of the requester, it can provide the resource if the requester is trusted by someone whom the server trusts.

However, in these days, these kinds of methods are used together in order to make the system more secure. For example, a customer wants to see a movie on a Web site. The Web site checks the identity of the customer, and then it checks whether he/she is allowed to see that movie or not. If the customer can see the movie, the system checks whether the customer does any illegal act to the contents. An attempt to unify all the methods is usage control (UCON) [2]. UCON is the conceptual and unified framework for these three efforts [3]. UCON can be applied to solve existing authorization problems and support finer-grained control than any other methods [4].

In the business model, there are some cases in which the consent of customer, originator, and any other concerned person must be resolved. For example, a patient's consent is needed when someone uses the information of the patient's disease in the healthcare system. Another example is a parents' consent when a child purchases some digital contents on the Web. Clarke introduced the e-Consent concept for proper control of these cases [5]. E-Consent handles the consent in the Web environment by using consent object. In the e-business, trust between customer and provider is an important issue. However, behind the Internet, how could the customer be convinced that the provider does not perform illegal act to the customer? In order to establish the trust between a customer and a provider, we should have a system that whenever the provider carries out an action that changes the customer's benefit, the customer's consent should be confirmed. Therefore, consent is important consideration in the trustful e-business.

In this paper, we propose a novel method for consent handling in UCON model. Currently the concept of consent does not provided by existing methods including the UCON. We consider the consent handling as the authorization problem. Therefore we modify the authorization model of UCON. We describe how to administrate the modified authorization model in UCON. The proposed method enhances the security of the UCON model since this consent provides another level of access control. This method extends the coverage of the UCON model in security area.

This paper is organized as follows. We describe the consent and UCON model more detail in section 2. In section 3, we explain how to change existing UCON model and its application. This is followed by the administrative issue on the proposed method in section 4. Section 5 concludes.

2 Backgrounds

2.1 Consents in E-business

The definition of consent is compliance in what is done, or approval of what is proposed by another. The consent could be used in many business models. We discuss three example cases. The first case is when a company uses customer's

information for the purpose of advertising the company, not for product shopping. The company must acquire the consent of the customer. The second case is in the hospital. When a patient wants to donate his/her organs or body, the consent of patient and guardian should be stated clearly. The third case is as follows; as the stockholder's consent, a company decides whether the annual report and proxy materials send to a stockholder via e-mail or not.

The consideration of the consent has been practiced in the law for long time. For the proper management of this, concept on the Web, e-Consent should consider the following characteristic of consent [6];

- Express consent: the concerned party in interests must save the proof of consent to support non-repudiation security requirements.
- Implied consent: if an individual's action responsibly causes another party to interpret it as having been granted, then the action is an implied consent. In this case, since another party infers the consent from the context, we can call it *inferred consent*.
- Denial of consent: in some cases, there exists a regulation that consent exists unless a person concerned does not expressively repeal, this nullity must be available to a person concerned.
- Informed consent: consent must include following information; what action, to whom, for what purpose, and for how long.
- Freely-given consent: the consent may be inducible to a person concerned, not by compulsion.
- Revocability and variability: the consent can be revoked and be changed by the consenter.

E-Consent is a method to apply consent to the cyber-world. Based on the characteristic of the consent, the system makes corresponding consent object. This object supports all the basic security services such as confidentiality, integrity, authenticity, and non-repudiation. To do so, many security features, such as encryption/decryption algorithm, digital signature, authentication methods, virtual private networks, and XML are used [5].

2.2 Usage Control (UCON)

The purpose of UCON is to unify all the authorization methods and to provide a new vision of access control for information and network security [2]. For example, in the case of selling e-book, before the decision of purchase is made, access control or trust management should be applied during a set of consumer's action, such as previewing the book and reading some comment for the book. After buying the e-book, whenever the consumer uses it, the digital rights management (DRM) should check whether he/she violates the regulation on the license. In these days, the fine-grained control for digital contents usage is needed [3]. For example, if a person purchases a voucher that he/she reads a book in ten days, existing method checks how many days elapsed from purchasing it. Moreover, the business model needs to check the accumulative time whenever he/she actually reads the book.

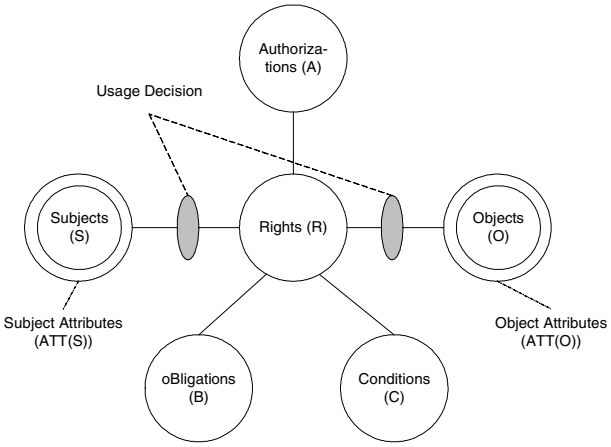


Fig. 1. The relationships of components of UCON model

Indeed, usage control has been researched on digital right management (DRM) and account management of the lined and mobile network [7], [8], and [9]. Park et al. proposed the ABC (Authorization, oBligation, and Condition) model for usage control to generalize and unify them [3].This model employs six components such as subject, object, authorization, obligation, condition, and right. Subject has three types such as consumer, provider, and identifee. The identifee is a subject that individual information such as credit card number is included in the contents. With these components, ABC model can support usage rules between subjects depending on the business context. The three usage rules are authorization, obligation, and condition. Fig. 1 shows the relationship of the components [3]. The model employs three predicate such as *pre*, *ongoing*, and *post* to control the usage transaction between subjects. With these predicate, UCON controls the usage of information and resource.

Fig. 2 shows the administrative view of the UCON [3]. This explains how the subjects control the usage of objects and how they interacts each other. There may be a set of provider related to an object. Such case is denoted by *serial usage controls*. On the other hand, there may be usage of an object that includes individual information of corresponding multiple subjects. Such case is denoted by *parallel usage control*.

3 Consent Handling Method in UCON

In UCON model, the consent is the concept which is opposite to the oBligation rule in some ways. While the obligation is obeyed by the customer, the consent is observed by the provider. For example, whenever a customer c_1 reads an object e-book o_1, c_1 must conform to the rule that the customer do not distribute this e-book by any illegal ways. This is the obligation. When the provider p_1 wants to distribute the e-book via the mobile network, p_1 must acquire the originator's

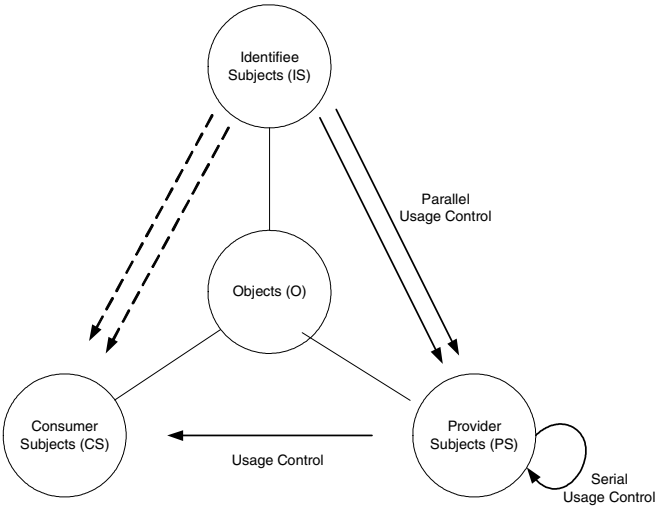


Fig. 2. The administrative view of the UCON model

agreement. This is the consent. Moreover, different from the obligation, the consent is considered at the authorization time. Once the provider gets the right related to his/her work, the provider performs the work unless the consent or authorization rules are changed.

With these considerations, we modify the authorization model of the UCON. The ABC core model for UCON employs two authorization model such as *pre-authorization* and *ongoing-authorization*. The predicate *pre* means the time before using the object and *ongoing* means the time during using the object [3]. According to the predicate, the *pre-authorization* is the model that checks the requester is allowed to use the object, while the *ongoing-authorization* is the model that terminates the usage of object when the requester's attributes, object's attributes, or the rights are changed.

To apply the consent at the authorization time, the necessity of consent should be decided. In order to check the necessity, the method checks the policy with the object's attributes and related rights. The necessity of the consent should follow the security policy, and the policy should conform to the electronic commerce regulations. After deciding the necessity, if the consent is needed, proposed method checks the consent is performed. According to the informed consent characteristic, consent must contain the information about (1) what action, (2) to whom, (3) for what purpose, and (4) for how long. Therefore, in this step, we can check whether the consent for the object requester is performed with the requester's attributes, object's attributes, and related rights. If the consent is performed and the authorization result is true, the requested object is provided to the requester.

In some cases, the authorization decision should be made continuously or repeatedly while using the same object. Accordingly, whenever decision is made,

the checking process of the consent is needed. By the same process, the consent checking step is also performed in the ongoing-authorization model.

The formalized definition of the modified model is as follows;

1. Pre-authorization
   - Subjects S, Objects O, Rights R, attributes of a subject $ATT(s)$, attributes of an object $ATT(o)$
   - Decision making on the authorization: $preA(ATT(s), ATT(o), r)$
   - Function that checks the necessity of consent: $needConsent(ATT(o), r)$
   - Function that checks whether related subject consent or not: $Consented(ATT(s), ATT(o), r)$
   - If $needConsent(ATT(o), r) \Rightarrow true$, then $allowed(s, o, r)$ $\Leftarrow preA(ATT(s), ATT(o), r) \bigcap Consented(ATT(s), ATT(o), r)$ Otherwise, $allowed(s, o, r) \Leftarrow preA(ATT(s), ATT(o), r)$
2. Ongoing-authorization
   - Subjects S, Objects O, Rights R, attributes of a subject $ATT(s)$, attributes of an object $ATT(o)$ are not changed from pre-authorization.
   - Decision making on the authorization: $preA(ATT(s), ATT(o), r)$
   - Function that checks the necessity of consent: $needConsent(ATT(o), r)$
   - Function that checks whether related subject consent or not: $Consented(ATT(s), ATT(o), r)$
   - Suppose that $allowed(s, o, r) \Rightarrow true$, if $needConsent(ATT(o), r) \Rightarrow true$, then $stopped(s, o, r)$ $\Leftarrow \neg onA(ATT(s), ATT(o), r) \bigcap Consented(ATT(s), ATT(o), r)$ Otherwise, $stopped(s, o, r) \Leftarrow \neg onA(ATT(s), ATT(o), r)$

4 Consent in UCON Administration

With the proposed model, we describe how to administrate the consent handling in UCON model. To apply the consent to UCON model, we modify the administrative view of existing UCON model. Fig. 3 shows the administrative view of proposed method. To simplify Fig. 3, we remove the parallel usage control arrows from the original administrative model, though the proposed model also supports it. There are three kinds of subject such as consumer, provider, and identifiee. A consumer subject is the end-user of contents in a supply chain. A provider subject is an originator of an object or any third party who provide the object. An identifiee subject is the subject whose individually identifiable information is included in the object. Each subject has their own rights to the object, and the usage of consumer is controlled by the provider. If there are several provider subject in a supply chain, the control of customer's usage is affected

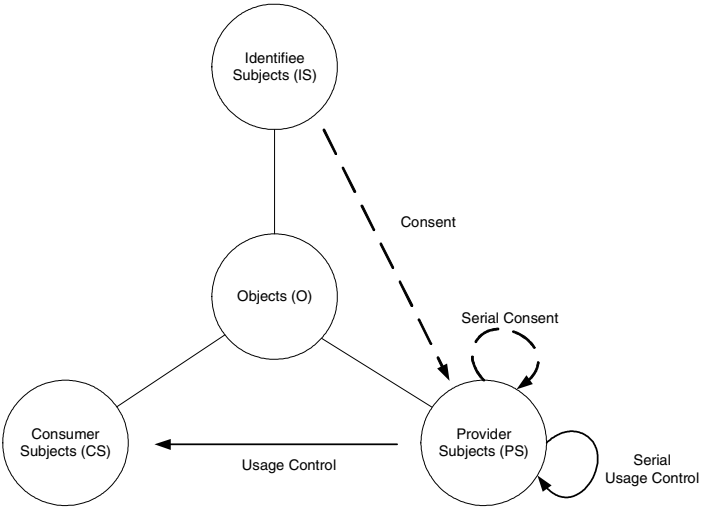


Fig. 3. The administrative view of consent handling in UCON model

by the all the related provider. For example, if a provider subject p_1 contracts the usage of an object o1 with an originator p_2 of o_1, then p_1 distributes a new object o_2 that includes the object o_1. In this case, p_1's ability on usage of o_1 is limited by the p_2 since p_1 is considered as a consumer of o_1. This usage control is denoted as serial usage control. If a customer c_1 purchase the object o_1, the c_1's usage of o_1 is controlled by p_1.

There are two dashed arrows of consent handling such as from identifiee subject to provider subject and from a provider subject to other provider subject. The latter case is denoted as serial consent. These arrows represent flow of consent. In the previous example, when the provider p_1 wants to modify the object o_2, the originator p_2 must consent to modifying the object. In this example, there is a consent flow between originator p_2 and provider p_1. The detailed descriptions are in the next section.

4.1 IS-to-IS Consent

This is the general case of consent handling for the privacy. In this case, an IS (identifiee subject) i_1 sends the consent to a PS (provider subject) p_1, and then provider can acquire the right on usage of object o_1 that includes private information such as phone number, address, and e-mail address. In Fig. 3, the arrow from IS to PS shows that the identifiee's consent is sent to the provider.

Here, we show some examples for this case of consent handling. A customer c_1 signs up for email account to a portal site such as yahoo. If the portal site p_1 contracts with an Internet shopping mall p_2 to provide a new shopping product, then p_2 uses the customer c_1's information such as e-mail and address. To do this, c_1's consent must be needed. In this case, the customer c_1 is considered as

the identifiee subject, and the portal site p_1 is regarded as the provider subject. The Internet shopping mall p_2 is considered as the customer subject. Lastly the database containing the c_1's information is the object.

Another example is a patient's information in a hospital. Some patients who are afflicted with the incurable disease do not want to be known to other people about their disease. Therefore, only consented person is allowed to access their information on the health information system. Even though reqeuster is a doctor, if he/she does not have patient's consent, he/she cannot access patient's information.

4.2 PS-to-PS Consent

In this section, we describe serial consent. As in Fig. 4, serial consent is occurred between providers, especially between originator and contents provider. This is the case in which the originator's consent is required for the provider to use the objects. In this case, if the originator p_1 sends the consent to the provider p_2, then p_1 acquires the ability on controlling the usage of object o_1. When a customer c_1 requests an object o_1, the provider p_2 checks the consent of p_1.

For example, if a patient c_2 wants to donate his body for other incurable patient c_3, then the hospital cannot use his/her body for the research since c_2 does not consent to using his body in anatomy experiment. In this case, the patient c_2 is considered as the originator and original provider. The hospital is the provider subject, and the patient c_3 is the customer subject. The body of c_2 is regarded as the object. Another example is the case when the provider manufactures the original object. This example is already explained in the first paragraph of section 4.

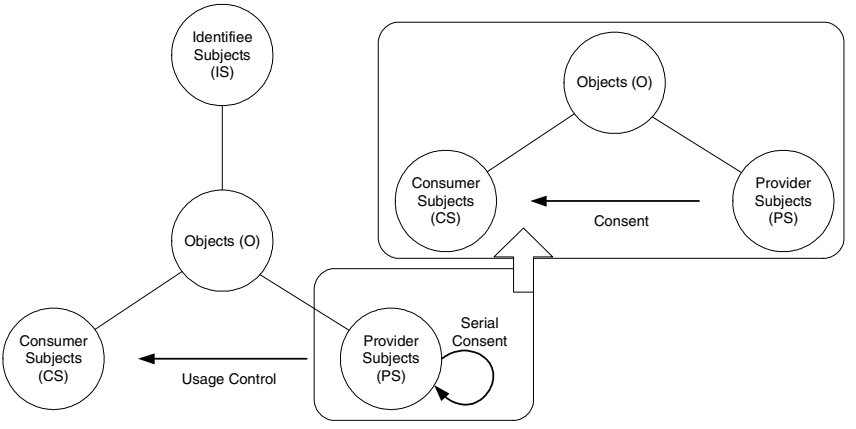


Fig. 4. Administrative view of the PS-to-PS consent handling

5 Conclusion

In many cases of e-commerce, e-government, and healthcare systems, the consent of related party is needed. However, any existing authorization method does not support this. In this paper, we proposed a novel method of consent handling in UCON model. The UCON model is suitable for the next generation access control. We modified the authorization model of UCON so that it can support the consent. The administrative issues of the proposed model are also discussed. The proposed method can extend the coverage of the UCON model in security area, and will enhance right of the provider and customer. It also provides a solution for trust relationships on e-Commerce and protection on individual privacy.

References

1. Li, N., Mitchell, J.C., Winsbrough, W.H.: Design of a Role-based Trust-Management Framework. Proceedings of the 2002 IEEE Symposium on Security and Privacy. Oakland, USA (2002)
2. Sandhu, R., Park, J.: Usage Control: A Vision for Next Generation Access Control. MMM-ACNS, St. Petersburg, Russia (2003)
3. Park, J., Sandhu, R.: The ABC Core Model for Usage Control: Integrating Authorizations, oBligations, and Conditions. ACM Transactions on Information and System Security, Vol. 7, No. 1 (2004) 1–47
4. Park, J., Sandhu, R.: Towards Usage Control Models: Beyond Traditional Access Control. Proceedings of seventh Symposium on Access Control Model and Technologies '02. Meonterey, California, USA (2002) 57–64
5. Clarke, R.: e-Consent: A Critical Element of Trust in e-Business. Proceedings of 15th Bled Electronic Commerce Conference, Bled, Russia (2002)
6. Clarke, R.: Consumer Consent in Electronic Health Data Exchange. Background Paper for Australian Government Department of Health and Aged Care (2002)
7. Xu, C., Zhu, Y., Feng, D.D.: Content Protection and Usage Control for Digital Music. Proceedings of the First International Conference on WEB Delivering of Music. Florence, Italy (2001) 43–51
8. Aboba,B., Arkko, J., Harrington, D.: Introduction to Accounting Management. RFC2975 (2000) URL: http://www.ietf.org/rfc/rfc2975.txt
9. Tashiro, S.: Capability-Based Usage Control Scheme for Network-Transferred Objects. INET97. Kuala Lumpur, Malaysia (1997)

No Trade under Rational Expectations in Economy

A Multi-modal Logic Approach

Takashi Matsuhisa*

Department of Liberal Arts and Sciences, Ibaraki National College of Technology
Nakane 866, Hitachinaka-shi, Ibaraki 312-8508, Japan.
mathisa@ge.ibaraki-ct.ac.jp

Dedicated to Nobuyoshi Motohashi on the occasion of his 60th birthday

Abstract. We investigate a pure exchange economy under uncertainty with emphasis on the logical point of view; the traders are assumed to have a multi-modal logic with non-partition information structure. We propose a pure exchange economy $\mathcal{E}^{KT}$ for the multi-modal logic **KT**, and extend the notion of rational expectations equilibrium for the economy. We characterize welfare under the generalized rational expectations equilibrium, and we show the no trade theorem: If the initial endowment allocation is ex-ante Pareto optimal then there exists no other rational expectations equilibrium for any price in the economy.

Keywords: No trade theorem, Distributed knowledge, Multi-modal logics, Exchange economy under uncertainty, Rational expectations equilibrium.
AMS 2000 Mathematics Subject Classification: Primary 91B50 Secondary 03B45, 91B60.
Journal of Economic Literature Classification: D51, D84, D52, C72.

1 Introduction

This article relates economies and distributed knowledge. The purposes are two points: First to establish the fundamental theorem for welfare in the pure exchange economy for the multi-modal logic **KT** by which the traders use making their decision. Secondly to prove

No trade theorem. *In a pure exchange economy under generalized information, assume that the traders have the multi-modal logic* **KT** *and they are risk averse. If the initial endowment are ex-ante Pareto optimal then there exists no other rational expectations equilibrium for any price in the economy.*

* Partially supported by the Grant-in-Aid for Scientific Research(C)(2)(No.14540145) in the Japan Society for the Promotion of Sciences.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 915–925, 2004.
© Springer-Verlag Berlin Heidelberg 2004

More recently, researchers in economics, AI, and computer science become entertained lively concerns about relationships between knowledge and actions. At what point does an economic agent (a trader) sufficiently know to stop gathering information and make decisions? At the heart of any analysis of such situations as a conversation, a bargaining session or a protocol run by processes is the interaction between agents (traders). An agent (a trader) in a group must take into account not only events that have occurred in the world but also the knowledge of the other agents in the group. The most interest to us is the emphasis on the considering the situation involving the distributed knowledge of multi-agents rather than of just a single agent.

Let us consider a pure exchange economy under uncertainty. Many authors have shown that there can be no trade in rational expectations (e.g., Kreps [6], Milgrom and Stokey [10], Geanakoplos [5], Morris [11] and others). The serious limitation of the analysis in these researches is its use of the 'partition' structure by which the traders receive information. It is obtained if each trader t's possibility operator $P_t : \Omega \to 2^{\Omega}$ assigning to each state ω in a state space Ω the information set $P_t(\omega)$ that t possesses in ω is reflexive, transitive and symmetric. This entails t's knowledge operator $K_t : 2^{\Omega} \to 2^{\Omega}$ that satisfies 'Truth' axiom **T**: $K_t(E) \subseteq E$ (what is known is true), the 'positive introspection' axiom **4**: $K_t(E) \subseteq K_t(K_t(E))$ (we know what we do) and the 'negative introspection' axiom **5**: $\Omega \setminus K_t(E) \subseteq K_t(\Omega \setminus K_t(E))$ (we know what we do not know). One of these requirements, symmetry (or the equivalent axiom **5**), is indeed so strong that describes the hyper-rationality of traders, and thus it is particularly objectionable. Dropping such assumptions can potentially yield important results in a world with imperfectly Bayesian agents.

The idea has been performed in different settings.[1] Among other things Geanakoplos [5] showed the no trade theorem in the extended rational expectations equilibrium under the assumption that the information structure is reflexive, transitive and *nested*. The condition 'nestedness' is interpreted as a requisite on the 'memory' of the traders. However all of the above mentioned researches have been lacked the logics representing the traders' knowledge. Although the partition structure is interpreted as the Kripke semantics for the modal logic **S5**,[2] the economy has not been investigated from the epistemic logical point of view. Matsuhisa [7] proposes the extended notion of the rational expectations equilibrium in the pure exchange economy for the multi-modal logic **KT**, and shows the existence theorem of the equilibrium. This article is a continuation of it.

The stage is set by the following: Suppose that the traders have the multi-agent modal logic **KT**: It is an extension of the propositional logic with traders' modal operators requiring only the axiom (T) "each trader does not know a sentence whenever it is not true." The logic has non-partition information structures as its semantics, each of which gives an interpretation of the logic. A trader receives information about the states of the world representable by the information

[1] The references cited in Fudenberg and Tirole [4], footnote 3, p.543

[2] C.f.: Chellas [2], Fagin, Halpern et al [3].

structure, and he/she has own utility function which is measurable but is not assumed that they know the function completely. A trading process takes place where the traders try to maximize their expected utilities. In this set up the no trade theorem says that equilibrium trade is null if the contingent commodities are ex-ante Pareto-optimally allocated.

This article is organized as follows: In Section 2 we present the multi-modal logic **KT** and its finite model property. Further we introduce the notion "economy for **KT**" and a generalized notion of rational expectations equilibrium. We note the existence theorem of rational expectations equilibrium. Section 3 gives the main theorems. In Subsection 3.1 we show the extended fundamental theorem for welfare in the economy for **KT**. This theorem plays an essential role in the proof of the no trade theorem. In Subsection 3.2 we give the explicit statement of the no trade theorem and the proof. Finally we conclude by giving some remarks about the assumptions of the theorem.

2 Pure Exchange Economy for Multi-modal Logic

2.1 Logic of Knowledge KT

Let T be a set of n *traders* $\{1, 2, 3, \ldots, t, \ldots, n\}$. Let us consider *multi-modal logics* for traders T as follows: The *sentences* of the language form the least set containing each *atomic* sentence $\mathbf{P}_m (m = 0, 1, 2, \ldots)$ and closed under the following operations:

- nullary operators for *falsity* $\bot$ and for *truth* $\top$;
- unary and binary syntactic operations for *negation* $\neg$, *conditionality* $\to$ and *conjunction* $\wedge$, respectively;
- unary operation for *modality* k_t with $t \in T$.

Other such operations are defined in terms of those in usual ways. The intended interpretation of $\mathrm{k}_t\varphi$ is the sentence that 'trader t knows a sentence φ.'

By a *multi-modal logic* we mean a set L of sentences containing all truth-functional tautologies and closed under both substitution and modus ponens. A multi-modal logic L' is an *extension* of L if $L \subseteq L'$. A sentence φ in L is said to be a *theorem* of L (or *provable* in L), written by $\vdash_L \varphi$. Other proof-theoretical notions such as *L-deducibility, L-consistency, L-maximality* are defined in usual ways. (See, Chellas [2].)

A *normal system* of traders' knowledge is a multi-modal logic L closed under the rule of inference $(\mathrm{RE_k})$ and containing the schema (N), (K), and (T): For every $t \in T$,

$$(\mathrm{RE_k}) \quad \frac{\varphi \longleftrightarrow \psi}{\mathrm{k}_t\varphi \longleftrightarrow \mathrm{k}_t\psi} \qquad (\mathrm{N}) \quad \mathrm{k}_t\top;$$

$$(\mathrm{K}) \quad \mathrm{k}_t(\varphi\wedge\psi) \longleftrightarrow (\mathrm{k}_t\varphi\wedge\mathrm{k}_t\psi); \qquad (\mathrm{T}) \quad \mathrm{k}_t\varphi \longrightarrow \varphi.$$

Definition 1. The *multi-modal logic* **KT** is the minimal system of trades' knowledge.

2.2 Information and Knowledge[3]

Trader t's information structure is a couple $\langle \Omega, P_t \rangle$, in which Ω is a non-empty set called a *state-space* whose elements are called *states* and P_t is a mapping of Ω into 2^{Ω}. It is said to be *reflexive* if

Ref $\quad \omega \in P_t(\omega) \quad$ for every $\omega \in \Omega$,

and it is said to be *transitive* if

Trn $\quad \xi \in P_t(\omega)$ implies $P_t(\xi) \subseteq P_t(\omega)$ for any $\xi, \omega \in \Omega$.

An *information structure* is a structure $\langle \Omega, (P_t)_{t \in T} \rangle$ where $\langle \Omega, P_t \rangle$ is t's information structure and Ω is common for all traders. It is called an *RT-information structure* if each P_t is reflexive and transitive.

Given our interpretation, a trader t for whom $P_t(\omega) \subseteq E$ knows, in the state ω, that some state in the event E has occurred. In this case we say that at the state ω the trader t knows E. i's *knowledge operator* K_t on 2^{Ω} is defined by $K_t(E) = \{\omega \in \Omega | P_t(\omega) \subseteq E\}$. The set $P_t(\omega)$ will be interpreted as the set of all the states of nature that t knows to be possible at ω, and $K_t E$ will be interpreted as the set of states of nature for which t knows E to be possible. We will therefore call P_t t's *possibility operator* and also will call $P_t(\omega)$ t's *possibility set* at ω. A possibility operator P_t is determined by the knowledge operator K_t such as $P_t(\omega) = \bigcap_{K_t E \ni \omega} E$.

A *partition* information structure is an *RT*-information structure $\langle \Omega, (P_t)_{t \in T} \rangle$ with the additional condition: For each $t \in T$ and every $\omega \in \Omega$,

Sym $\quad \xi \in P_t(\omega)$ implies $P_t(\xi) \ni \omega$.

2.3 Truth

A *model on* an information structure $\langle \Omega, (P_t)_{t \in T} \rangle$ is a triple $\mathcal{M} = \langle \Omega, (P_t)_{t \in T}, V \rangle$, in which a mapping V assigns either **true** or **false** to every $\omega \in \Omega$ and to every atomic sentence $\mathbf{P}_m$. The model $\mathcal{M}$ is called *finite* if Ω is a finite set.

Definition 2. By $\models_{\omega}^{\mathcal{M}} \varphi$ we mean that a sentence φ is *true* at a state ω in a model $\mathcal{M}$. *Truth at a state* ω in $\mathcal{M}$ is inductively defined as follows:

1. $\models_{\omega}^{\mathcal{M}} \mathbf{P}_m$ if and only if $V(\omega, \mathbf{P}_m) = \mathbf{true}$, for $m = 0, 1, 2, \ldots$;
2. $\models_{\omega}^{\mathcal{M}} \top$, and not $\models_{\omega}^{\mathcal{M}} \bot$;
3. $\models_{\omega}^{\mathcal{M}} \neg\varphi$ if and only if not $\models_{\omega}^{\mathcal{M}} \varphi$;
4. $\models_{\omega}^{\mathcal{M}} \varphi \longrightarrow \psi$ if and only if $\models_{\omega}^{\mathcal{M}} \varphi$ implies $\models_{\omega}^{\mathcal{M}} \psi$;
5. $\models_{\omega}^{\mathcal{M}} \varphi \wedge \psi$ if and only if $\models_{\omega}^{\mathcal{M}} \varphi$ and $\models_{\omega}^{\mathcal{M}} \psi$;

[3] See Fagin, Halpern et al [3].

6. $\models^{\mathcal{M}}_{\omega} \mathrm{k}_t\varphi$ if and only if $P_t(\omega) \subseteq ||\varphi||^{\mathcal{M}}$, for $t \in T$;

Where $||\varphi||^{\mathcal{M}}$ denotes the set of all the states in $\mathcal{M}$ at which φ is true; this is called the *truth set of* φ. We say that a sentence φ is *true in the model* $\mathcal{M}$ and write $\models^{\mathcal{M}} \varphi$ if $\models^{\mathcal{M}}_{\omega} \varphi$ for every state ω in $\mathcal{M}$. A sentence is said to be *valid in* an information structure if it is true in every model on the information structure.

2.4 Finite Model Property

Let Σ be a set of sentences. We say that $\mathcal{M}$ is a *model for* Σ if every member of Σ is true in $\mathcal{M}$. An information structure is said to be *for* Σ if every member of Σ is valid in it. Let $\mathbf{R}$ be the class of all models on any *reflexive* information structure. A multi-modal logic L is *sound with respect to* $\mathbf{R}$ if every member of $\mathbf{R}$ is a model on an information structure for L. It is *complete with respect to* $\mathbf{R}$ if every sentence valid in all members of $\mathbf{R}$ is a theorem of L. We say that L is *determined by* $\mathbf{R}$ if L is sound and complete with respect to $\mathbf{R}$.

A multi-modal logic L is said to have the *finite model property* if it is determined by the class of all finite models in $\mathbf{R}$.

Theorem 1. *The multi-modal logic* **KT** *has the finite model property.*

Proof. Is given by the same way as described in Chellas [2]. □

From now on we consider the structure $\langle \Omega, (P_t)_{t\in T}, V\rangle$ as a finite model for **KT**.

2.5 Economy for Logic KT

Let Ω be a non-empty *finite* state space and let 2^{Ω} denote the field of all subsets of Ω. Each member of 2^{Ω} is called an *event*.

A *pure exchange economy under uncertainty* is a tuple $\langle T, \Omega, \mathbf{e}, (U_t)_{t\in T}, (\pi_t)_{t\in T}\rangle$ consisting of the following structure and interpretations: There are l commodities in each state of the state space Ω, and it is assumed that the consumption set of trader t is $\mathbb{R}^l_+$;

- $\mathbf{e}(t,\cdot) : T \times \Omega \to \mathbb{R}^l_+$ is t's *initial endowment*;
- $U_t : \mathbb{R}^l_+ \times \Omega \to \mathbb{R}$ is t's von-Neumann and Morgenstern utility function;
- π_t is a subjective prior on Ω for a trader $t \in T$.

For simplicity it is assumed that (Ω, π_t) is a finite probability space with π_t *full support*[4]for all $t \in T$.

Definition 3. A *pure exchange economy for logic* **KT** is a structure $\mathcal{E}^{KT} = \langle \mathcal{E}, (P_t)_{t\in T}, V\rangle$, in which $\mathcal{E}$ is a pure exchange economy such that $\langle \Omega, (P_t)_{t\in T}, V\rangle$ is a finite model for the logic **KT**. Furthermore it is called an economy *under RT-information structure* if each P_t is reflexive and transitive.

[4] I.e., $\pi_t(\omega) \neq 0$ for every $\omega \in \Omega$.

Remark 1. An economy under asymmetric information is an economy $\mathcal{E}^{KT}$ under partition information structure (i.e., each P_t satisfies the three conditions **Ref**, **Trn** and **Sym**.)

We denote by $\mathcal{F}_t$ the field generated by $\{P_t(\omega) \mid \omega \in \Omega\}$ and by $\mathcal{F}$ the join of all $\mathcal{F}_t (t \in T)$; i.e. $\mathcal{F} = \vee_{t\in T}\mathcal{F}_t$. We denote by $\{A(\omega) \mid \omega \in \Omega\}$ the set of all atoms $A(\omega)$ containing ω of the field $\mathcal{F} = \vee_{t\in T}\mathcal{F}_t$.

We shall often refer to the following conditions: For every $t \in T$,

A-1 The function $\mathbf{e}(t,\cdot)$ is $\mathcal{F}_t$-measurable with $\sum_{t\in T}\mathbf{e}(t,\omega) > 0$ for all $\omega \in \Omega$.

A-2 For each $x \in \mathbb{R}^l_+$, the function $U_t(x,\cdot)$ is $\mathcal{F}_t$-measurable.

A-3 For each $\omega \in \Omega$, the function $U_t(\cdot,\omega)$ is continuous, increasing, strictly quasi-concave and *non-saturated*[5] on $\mathbb{R}^l_+$.

Remark 2. The condition **A-3** implies that $U_t(\cdot,\omega)$ is strictly increasing on $\mathbb{R}^l_+$.

An *assignment* $\mathbf{x}$ is a mapping from $T \times \Omega$ into $\mathbb{R}^l_+$ such that for every $\omega \in \Omega$ and for each $t \in T$, the function $\mathbf{x}(t,\cdot)$ is at most $\mathcal{F}$-measurable. We denote by $\mathcal{A}ss(\mathcal{E}^{KT})$ the set of all assignments for the economy $\mathcal{E}^{KT}$. By an *allocation* we mean an assignment $\mathbf{a}$ such that for every $\omega \in \Omega$, $\sum_{t\in T}\mathbf{a}(t,\omega) \leq \sum_{t\in T}\mathbf{e}(t,\omega)$. We denote by $\mathcal{A}lc(\mathcal{E}^{KT})$ the set of all allocations, and for each $t \in T$ we denote by $\mathcal{A}lc(\mathcal{E}^{KT})_t$ the set of all the functions $\mathbf{a}(t,\cdot)$ for $\mathbf{a} \in \mathcal{A}lc(\mathcal{E}^{KT})$.

By the *ex-ante* expectation we mean

$$\mathbf{E}_t[U_t(\mathbf{x}(t,\cdot)] := \sum_{\omega\in\Omega} U_t(\mathbf{x}(t,\omega),\omega)\pi_t(\omega)$$

for each $\mathbf{x} \in \mathcal{A}ss(\mathcal{E}^{KT})$. i's *interim* expectation is defined by

$$\mathbf{E}_t[U_t(\mathbf{x}(t,\cdot)|P_t](\omega) := \sum_{\xi\in\Omega} U_t(\mathbf{x}(t,\xi),\xi)\pi_t(\xi|P_t(\omega)).$$

Definition 4. An allocation $\mathbf{x}$ in an economy $\mathcal{E}^{KT}$ is said to be *ex-ante Pareto-optimal* if there is no allocation $\mathbf{a}$ such that $\mathbf{E}_t[U_t(\mathbf{a}(t,\cdot)] \geq \mathbf{E}_t[U_t(\mathbf{x}(t,\cdot)]$ for all $t \in T$, with one inequality strict.

2.6 Rational Expectations Equilibrium

A *price system* is a non-zero $\mathcal{F}$-measurable function $p : \Omega \to \mathbb{R}^l_+$. We denote by $\sigma(p)$ the smallest σ-field containing the partition induced by p: The component of the partition in ω is given by $\Delta(p)(\omega) = \{\xi \in \Omega \mid p(\xi) = p(\omega)\}$, called the atom containing ω of the field $\sigma(p)$. The *budget set* of a trader t at a state ω for a price system p is defined by

$$B_t(\omega,p) := \{\ x \in \mathbb{R}^l_+ \mid p(\omega)\cdot x \leq p(\omega)\cdot \mathbf{e}(t,\omega)\ \}.$$

[5] That is, for any $x \in \mathbb{R}^l_+$ there exists an $x' \in \mathbb{R}^l_+$ such that $U_t(x',\omega) > U_t(x,\omega)$.

Let $\Delta(p) \cap P_t : \Omega \to 2^{\Omega}$ be defined by $(\Delta(p) \cap P_t)(\omega) := \Delta(p)(\omega) \cap P_t(\omega)$; it is plainly observed that the mapping $\Delta(p) \cap P_t$ satisfies **Ref** and **Trn**. We denote by $\sigma(p) \vee \mathcal{F}_t$ the smallest σ-field containing both the fields $\sigma(p)$ and $\mathcal{F}_t$, and by $A_t(p)(\omega)$ the atom containing ω. It is noted that $A_t(p)(\omega) = (\Delta(p) \cap A_t)(\omega)$.

We shall give the extended notion of rational expectations equilibrium for an economy $\mathcal{E}^{KT}$.

Definition 5. A *rational expectations equilibrium* for an economy $\mathcal{E}^{KT}$ is a pair $(p, \mathbf{x})$, in which p is a price system and $\mathbf{x}$ is an allocation satisfying the following conditions: For all $t \in T$ and for every $\omega \in \Omega$,

RE 1 $\mathbf{x}(t, \cdot)$ is $\sigma(p) \vee \mathcal{F}_t$-measurable.

RE 2 $\mathbf{x}(t, \omega) \in B_t(\omega, p)$.

RE 3 If $\mathbf{y}(t, \cdot) : \Omega \to \mathbb{R}^l_+$ is $\sigma(p) \vee \mathcal{F}_t$-measurable with $\mathbf{y}(t, \omega) \in B_t(\omega, p)$ for all $\omega \in \Omega$, then $\mathbf{E}_t[U_t(\mathbf{x}(t, \cdot))|\Delta(p) \cap P_t](\omega) \geq \mathbf{E}_t[U_t(\mathbf{y}(t, \cdot))|\Delta(p) \cap P_t](\omega)$ pointwise on Ω.

RE 4 $\sum_{t \in T} \mathbf{x}(t, \omega) = \sum_{t \in T} \mathbf{e}(t, \omega)$.

The allocation $\mathbf{x}$ in $\mathcal{E}^{KT}$ is called a *rational expectations equilibrium allocation.*

We denote by $RE(\mathcal{E}^{KT})$ the set of all the rational expectations equilibria of the economy $\mathcal{E}^{KT}$, and denote by $\mathcal{R}(\mathcal{E}^{KT})$ the set of all the rational expectations equilibrium allocations for the economy.

Theorem 2. *Let $\mathcal{E}^{KT}$ be a pure exchange economy for logic* **KT** *satisfying the conditions* **A-1**, **A-2** *and* **A-3**. *Then there exists a rational expectations equilibrium for the economy; i.e., $RE(\mathcal{E}^{KT}) \neq \emptyset$.*

Proof. Is given in Matsuhisa [7]. □

3 Main Theorems

3.1 Fundamental Theorem for Welfare Economics

We shall characterize welfare under the generalized rational expectations equilibrium for an economy $\mathcal{E}^{KT}$ for logic **KT**.

Theorem 3. *Let $\mathcal{E}^{KT}$ be a pure exchange economy for logic* **KT** *satisfying the conditions* **A-1**, **A-2** *and* **A-3**. *An allocation is ex-ante Pareto optimal if and only if it is a rational expectations equilibrium allocation relative to some price system for $\mathcal{E}^{KT}$.*

Proof. Will be given in Subsection 3.3 below.

3.2 No Trade Theorem

We shall state the main theorem explicitly and prove it. The theorem is an extension of the no speculation theorem in Geanakoplos [5].

Theorem 4. *Let $\mathcal{E}^{KT}$ be a pure exchange economy for logic* **KT** *satisfying the conditions* **A-1**,**A-2** *and* **A-3**. *Suppose that the initial endowment* $\mathbf{e}$ *is ex-ante Pareto optimal in $\mathcal{E}^{KT}$. If $(p, \mathbf{x})$ is a rational expectations equilibrium for $\mathcal{E}^{KT}$ for some price system p then $\mathbf{x} = \mathbf{e}$.*

Before proceeding with we need notations. Let $\mathcal{E}^{KT}$ be a pure exchange economy $\langle T, \Omega, \mathbf{e}, (U_t)_{t\in T}, (\pi_t)_{t\in T}, (P_t)_{t\in T}, V\rangle$ for logic **KT**. We set by $\mathcal{E}^{KT}(\omega)$ the pure exchange economy $\langle T, (\mathbf{e}(t,\omega))_{t\in T}, (U_t(\cdot,\omega))_{t\in T}\rangle$ with complete information in a state $\omega \in \Omega$. We denote by $W(\mathcal{E}^{KT}(\omega))$ the set of all the competitive equilibria for $\mathcal{E}^{KT}(\omega)$, and by $\mathcal{W}(\mathcal{E}^{KT}(\omega))$ the set of all the competitive equilibrium allocations for $\mathcal{E}^{KT}(\omega)$.

Proof (of Theorem 4). Let $(p, \mathbf{x}) \in RE(\mathcal{E}^{KT})$. It follows from Theorem 3 that $\mathbf{x}$ is ex-ante Pareto optimal in $\mathcal{E}^{KT}$. Suppose to the contrary that $\mathbf{x} \neq \mathbf{e}$. Since $\mathbf{e}$ is ex-ante Pareto optimal in $\mathcal{E}^{KT}$ it can be observed that there exist a $s \in T$ such that $\mathbf{E}_s[U_s(\mathbf{e}(s,\cdot))] > \mathbf{E}_s[U_s(\mathbf{x}(s,\cdot))]$. Therefore, it can be plainly verified that for some $\omega_0 \in \Omega$, $U_s(\mathbf{e}(s,\omega_0),\omega_0) > U_s(\mathbf{x}(s,\omega_0),\omega_0)$. On the other hand, it follows from Proposition 1 below that $(p'(\omega_0), \mathbf{x}(\cdot,\omega_0)) \in W(\mathcal{E}^{KT}(\omega_0))$ for some price system p', and thus $U_s(\mathbf{x}(s,\omega_0),\omega_0) \geq U_s(\mathbf{e}(s,\omega_0),\omega_0)$, in contradiction. □

Proposition 1. *Let $\mathcal{E}^{KT}$ be the same in Theorem 4. The set of all rational expectations equilibrium allocations $\mathcal{R}(\mathcal{E}^{KT})$ coincides with the set of all the assignments* $\mathbf{x}$ *such that $\mathbf{x}(\cdot,\omega)$ is a competitive equilibrium allocation for the economy with complete information $\mathcal{E}^{KT}(\omega)$ for all $\omega \in \Omega$; i.e.,*

$$\mathcal{R}(\mathcal{E}^{KT}) = \{\mathbf{x} \in \mathcal{A}lc(\mathcal{E}^{KT}) \mid \textit{There is a price system } p \textit{ such that } (p(\omega), \mathbf{x}(\cdot,\omega)) \in W(\mathcal{E}^{KT}(\omega)) \textit{ for all } \omega \in \Omega\}.$$

Proof. Is given in Matsuhisa, Ishikawa and Hoshino [9]. □

3.3 Proof of Fundamental Theorem

Theorem 3 immediately follows from Propositions 2 and 3 below.

Proposition 2. *Let $\mathcal{E}^{KT}$ be the same in Theorem 3. If an allocation* $\mathbf{x}$ *is ex-ante Pareto optimal then it is a rational expectations equilibrium allocation relative to some price system.*

Proof. For each $\omega \in \Omega$ we denote

$$G(\omega) = \{ \int_T \mathbf{x}(t,\omega)d\mu - \int_T \mathbf{y}(t,\omega)d\mu \in \mathbb{R}^l_+ \mid \mathbf{y} \in \mathcal{A}ss(\mathcal{E}^K) \text{ and } U_t(\mathbf{y}(t,\omega),\omega) \geq U_t(\mathbf{x}(t,\omega),\omega) \text{ for almost all } t \in T\}.$$

First, we note that that $G(\omega)$ is convex and closed in $\mathbb{R}^l_+$ by the conditions **A-1**, **A-2** and **A-3**. It can be shown that

Claim 1: For each $\omega \in \Omega$ there exists $p^*(\omega) \in \mathbb{R}^l_+$ such that $p^*(\omega) \cdot v \leq 0$ for all $v \in G(\omega)$.

Proof of Claim 1: By the separation theorem[6] we can plainly observe that the assertion immediately follows from that $v \leq 0$ for all $v \in G(\omega)$: Suppose to the contrary that there exist $\omega_0 \in \Omega$ and $v_0 \in G(\omega_0)$ with $v_0 > 0$. Take an assignment $\mathbf{y}^0$ for $\mathcal{E}^{KT}$ such that for almost all t, $U_t(\mathbf{y}^0(t,\omega),\omega_0) \geq U_t(\mathbf{x}(t,\omega_0),\omega_0)$ and $v_0 = \int_T \mathbf{x}(t,\omega_0)d\mu - \int_T \mathbf{y}^0(t,\omega_0)d\mu$. Let $\mathbf{z}$ be the allocation defined by $\mathbf{z}(t,\xi) := \mathbf{y}^0(t,\omega_0) + \frac{v_0}{|T|}$ if $\xi \in A(\omega_0)$, and $\mathbf{z}(t,\xi) := \mathbf{x}(t,\xi)$ otherwise. It follows that for all $t \in T$,

$$\begin{aligned}\mathbf{E}_t[U_t(\mathbf{z})] &= \sum_{\xi \in A(\omega_0)} U_t(\mathbf{y}^0(t,\omega_0) + \frac{v_0}{|T|},\xi)\pi_t(\xi) + \sum_{\xi \in \Omega \setminus A(\omega_0)} U_t(\mathbf{x}(t,\xi),\xi)\pi_t(\xi) \\ &> \sum_{\xi \in A(\omega_0)} U_t(\mathbf{y}^0(t,\omega_0),\xi)\pi_t(\xi) + \sum_{\xi \in \Omega \setminus A(\omega_0)} U_t(\mathbf{x}(t,\xi),\xi)\pi_t(\xi) \\ &\geq \mathbf{E}_t[U_t(\mathbf{x})].\end{aligned}$$

This is in contradiction to which $\mathbf{x}$ is ex-ante Pareto optimal as required.

Secondly, let p be the price system defined as follows: We take a set of strictly positive numbers $\{k_\omega\}_{\omega \in \Omega}$ such that $k_\omega p^*(\omega) \neq k_\xi p^*(\xi)$ for any $\omega \neq \xi$. We define the price system p such that for each $\omega \in \Omega$ and for all $\xi \in A(\omega)$, $p(\xi) := k_\omega p^*(\omega)$. It can be observed that $\Delta(p)(\omega) = A(\omega)$. To conclude the proof we shall show

Claim 2: The pair $(p, \mathbf{x})$ is an expectations equilibrium for $\mathcal{E}^{KT}$.

Proof of Claim 2: We first note that for every $t \in T$ and for every $\omega \in \Omega$, $(\Delta(p) \cap P_t)(\omega) = \Delta(p)(\omega) = A(\omega)$. Therefore it follows from **A-3** that for every allocation $\mathbf{x}$,

$$\mathbf{E}_t[U_t(\mathbf{x}(t,\cdot))|(\Delta(p) \cap P_t)](\omega) = U_t(\mathbf{x}(t,\omega),\omega) \tag{1}$$

To prove Claim 2 it suffices to verify that $\mathbf{x}$ satisfies **RE 3**. Suppose to the contrary that there exists $s \in T$ with the two properties: (i) There is a $\sigma(p) \vee \mathcal{F}_s$-measurable function $\mathbf{y}(s,\cdot) : \Omega \to \mathbb{R}^l_+$ such that $\mathbf{y}(s,\omega) \in B_s(\omega,p)$ for all $\omega \in \Omega$; and (ii) $\mathbf{E}_s[U_s(\mathbf{y}(s,\cdot))|(\Delta(p) \cap P_s)](\omega_0) > \mathbf{E}_s[U_s(\mathbf{x}(s,\cdot)|(\Delta(p) \cap P_s)](\omega_0)$ for some $\omega_0 \in \Omega$. In view of Eq (1) it immediately follows from Property (ii) that $U_s(\mathbf{y}(s,\omega_0),\omega_0) > U_s(\mathbf{x}(s,\omega_0),\omega_0)$, and thus $\mathbf{y}(s,\omega_0) > \mathbf{x}(s,\omega_0)$ by **A-3**. Therefore we obtain that for all $s \in S$, $p(\omega_0) \cdot \mathbf{y}(s,\omega_0) > p(\omega_0) \cdot \mathbf{x}(s,\omega_0)$ in contradiction. □

Proposition 3. *Let $\mathcal{E}^{KT}$ be the same in Theorem 3. Then an allocation $\mathbf{x}$ is ex-ante Pareto optimal if it is a rational expectations equilibrium allocation relative to a price system.*

Proof. Let $(p, \mathbf{x})$ be a rational expectations equilibrium for $\mathcal{E}^{KT}$. It follows from Proposition 1 that $(p(\omega), \mathbf{x}(\cdot,\omega))$ is a competitive equilibrium for the economy

[6] C.f. Lemma 8, Chapter 4 in Arrow and Hahn [1].

with complete information $\mathcal{E}^{KT}(\omega)$ at each $\omega \in \Omega$. Therefore in viewing the standard fundamental theorem of welfare in the economy $\mathcal{E}^{KT}(\omega)$ with complete information, we can plainly observe that for all $\omega \in \Omega$, $\mathbf{x}(\cdot, \omega)$ is Pareto optimal in $\mathcal{E}^{KT}(\omega)$, and thus $\mathbf{x}$ is ex-ante Pareto optimal. □

4 Concluding Remarks

We shall give a remark about the auxiliary assumptions in results in this article. Could we prove the theorems under the generalized information structure removing out the reflexivity? The answer is no vein. If trader t's possibility operator does not satisfy **Ref** then his/her expectation with respect to a price cannot be defined at a state because it is possible that $\Delta(p)(\omega) \cap P_t(\omega) = \emptyset$ for some $\omega \in \Omega$. Could we prove the theorems without four conditions **A-1**, **A-2**, **A-3** and **A-3**? The answer is no again. The suppression of any of these assumptions renders the existence theorem of rational expectations equilibrium (Theorem 2) vulnerable to the discussion and the example proposed in Remarks 4.6 of Matsuhisa and Ishikawa [8].

It well ends related results: Matsuhisa and Ishikawa [8] extend the no trade theorem of Milgrom and Stokey into the economy under RT-information structure without the 'nestedness' condition in Geanakoplos [5] but with traders' rationality assumption about their expectations. Matsuhisa [7] presents the notion of ex-post core in the economy for logic **KT** equipped with non-atomic measure on the traders space T, and he establishes the core equivalence theorem based on Matsuhisa, Ishikawa and Hoshino [9]: The ex-post core in the economy for logic **KT** coincides with the set of all its rational expectations equilibria.

References

1. Arrow, K. J. and Hahn, F. H., 1971, *General competitive analysis* (North-Holland, Amsterdam, xii + 452pp.)
2. Chellas, B. F.: *Modal Logic: An introduction.* Cambridge University Press, Cambridge, London, New York, New Rochelle, Melbourne, Sydney (1980)
3. Fagin, R., Halpern, J.Y., Moses, Y., and Vardi, M.Y. *Reasoning about Knowledge.* The MIT Press, Cambridge, Massachusetts, London, England (1995)
4. Fudenberg, D. and Tirole, J.,*Game Theory.* MIT Press, Cambridge USA, xxii+579, 1991
5. Geanakoplos, J.: Game theory without partitions, and applications to speculation and consensus, Cowles Foundation Discussion Paper No.914 (1989) (Available in http://cowles.econ.yale.edu)
6. Kreps, D.: A note on fulfilled expectations equilibrium, Journal of Economic Theory **14** (1977) 32–44.
7. Matsuhisa, T.: Core equivalence in economy for modal logic, in P.M.A. Sloot, D. Abramson et al. (eds): Computational Science-ICCS 2003, Springer Lecture Notes in Computer Science, Vol. **2658** (2003) 74–83.
8. Matsuhisa, T. and Ishikawa, R.: Rational expectations can preclude trades. Preprint, Hitotsubashi Discussion Paper Series 2002-1 (2002) (Available in http://wakame.econ.hit-u.ac.jp/).

9. Matsuhisa, T., Ishikawa, R. and Hoshino, Y., 2002. Core equivalence in economy under generalized information. Working paper. Hitotsubashi Discussion Paper Series No.2002-12 (2002) (Available in http://wakame.econ.hit-u.ac.jp/)
10. Milgrom, P. and Stokey, N.: Information, trade and common knowledge, Journal of Economic Theory **26** (1982) 17–27.
11. Morris, S.: Trade with heterogeneous prior beliefs and asymmetric information, Econometrica **62** (1994) 1327–1347.

A New Approach for Numerical Identification of Optimal Exercise Curve

Chung-Ki Cho[1⋆], Sunbu Kang[2], Taekkeun Kim[2], and YongHoon Kwon[2⋆⋆]

[1] Department of Mathematics, Soonchunhyang University,
Asan 336-745, South Korea,
ckcho@sch.ac.kr,
[2] Department of Mathematics, Pohang University of Science and Technology,
Pohang 790-784, South Korea,
{sbkang, kz, ykwon}@postech.ac.kr

Abstract. This paper deals with American put options, which is modelled by a free boundary problem for a nonhomogeneous generalized Black-Scholes equation. We present a parameter estimation technique to compute the put option price as well as the optimal exercise curve. The forward problem of computing the put option price with a given parameter of the function space for the free boundary employs the upwind finite difference scheme. The inverse problem of minimizing the cost functional over that function space uses the Levenberg-Marquardt method. Numerical experiments show that the approximation scheme satisfies appropriate convergence properties. Our method can be applied to the case that the volatility is a function of time and asset variables.

1 Introduction

Being different from European options, American options can be exercised at any time up until the expiration date. So, the optimal exercise curve determining the best time for exercising an American option is one of the main concerns in mathematical finance.

No explicit closed-form formula for the optimal exercise curves is known yet, and hence there have been so many numerical studies on this subject, see, for example, [6,12,13] and references therein. In this paper, we propose a parameter estimation technique to compute the optimal exercise curve for the American put options.

It is observed by Merton [9] and McKean [8] that American option pricing problem is described by a homogeneous Black-Scholes equation, in which the free boundary is identified with the optimal exercise curve. It is noted by Jamshidian [4] that the free boundary problem can be regarded as a fixed boundary

⋆ The work of this author was supported by Korea Research Foundation Grant (KRF-2001-002-D00035)

⋆⋆ This paper is partially supported by Com^{2}MaC-KOSEF and by Postech Research Fund.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 926–934, 2004.
© Springer-Verlag Berlin Heidelberg 2004

problem for a nonhomogeneous Black-Scholes equation. Being motivated with this observation and further, we consider the coupled system (1) and (3) of the nonhomogeneous generalized Black-Scholes equation and the condition on the optimal exercise curve: to find a pair $(\mathrm{v}, \mathrm{s_p})$ satisfying

$$\frac{\partial \mathrm{v}}{\partial t} + \frac{1}{2}\sigma^2(t,s)s^2\frac{\partial^2 \mathrm{v}}{\partial s^2} + (r-d)s\frac{\partial \mathrm{v}}{\partial s} - r\mathrm{v} = F(t,s,\mathrm{s_p}), \tag{1}$$

in the domain $\{(t,s) \in (0,T)\times(0,S_{max})\}$, the terminal and boundary conditions

$$\mathrm{v}(T,s) = \max(E-s,0), \quad \mathrm{v}(t,0) = E, \quad \mathrm{v}(t,S_{max}) = 0, \tag{2}$$

and the (free boundary) condition

$$\mathrm{v}(t,\mathrm{s_p}(t)) = E - \mathrm{s_p}(t). \tag{3}$$

Here, $\mathrm{v}(t,s)$ is the put option price at time t and at asset price s, E the exercise price, T the expiration date, r the risk-free interest rate, σ the volatility of the underlying asset, and the curve $\{(t,s)|s = \mathrm{s_p}(t)\}$ the optimal exercise curve.

Note that the case of $F(t,s,\mathrm{s_p}) = (ds - rE)H(\mathrm{s_p}(t) - s)$ and $\sigma(t,s) =$ constant corresponds to the Jamshidian equation of the classical American put option problem [4], where H denotes the Heaviside function, except that the domain is $(0,T)\times(0,S_{max})$ instead of $(0,T)\times(0,\infty)$. The finite domain is taken for the computational feasibility, and it is reported in [7] that the choice $\mathrm{S}_{max} = 3E$ could be reasonably used.

Now we illustrate the problem (1)-(3) as a parameter estimation problem. Let Q be the set of all *possible optimal exercise curves* such that for each $\mathrm{q} \in \mathrm{Q}$, there exists a unique solution $\mathrm{p}(t,s;\mathrm{q})$ of the following terminal-boundary value problem

$$\frac{\partial \mathrm{p}}{\partial t} + \frac{1}{2}\sigma^2(t,s)s^2\frac{\partial^2 \mathrm{p}}{\partial s^2} + (r-d)s\frac{\partial \mathrm{p}}{\partial s} - r\mathrm{p} = F(t,s,\mathrm{q}), \tag{4}$$

$$\mathrm{p}(T,s) = \max(E-s,0), \quad \mathrm{p}(t,0) = E, \quad \mathrm{p}(t,S_{max}) = 0. \tag{5}$$

So, we get a map $\mathrm{q} \mapsto \mathrm{p}(\mathrm{q})$. Suppose $(\mathrm{v}, \mathrm{s_p})$ is a solution of (1)-(3). Then, it is clear that the solution p of (4)-(5) corresponding to $\mathrm{q} = \mathrm{s_p}$ is the very v, and hence

$$\mathrm{p}(t,\mathrm{s_p}(t);\mathrm{s_p}) = E - \mathrm{s_p}(t)$$

holds. Thus the optimal exercise curve $\mathrm{s_p}$ is a minimizer of the map

$$\mathrm{J} : \mathrm{Q} \to \mathbb{R}, \quad \mathrm{q} \mapsto ||\mathrm{p}(t,\mathrm{q}(t);\mathrm{q}) - E + \mathrm{q}(t)||,$$

where $||\cdot||$ denotes a norm on Q.

This optimization problem is nonlinear. We should solve (4)-(5) to calculate J. Furthermore, the problem is an infinite dimensional one, since the solution space $\{\mathrm{p}\}$ as well as the parameter space $\{\mathrm{q}\}$ are infinite dimensional function

spaces. Thus, we need to approximate $\mathrm{s_p}$ by a sequence of finite dimensional problems through the finite dimensional approximations of the solution spaces and parameter spaces.

This paper is organized as follows. In Section 2, we describe an upwind finite difference method for numerical solution p of (4)-(5). In Section 3, we address the finite dimensional approximation of our optimization problem for parameter estimation and discuss the suitable concept of convergence. Section 4 is devoted to numerical simulations. Finally, in Section 5, we give conclusions including some comments.

2 Forward Problem

In Section 2, we describe a numerical approximation scheme for (4)-(5). Throughout this section we assume that a function parameter $\mathrm{q} \in \mathrm{Q}$ is fixed.

Let N and M be given positive integers and let $k = T/N$ and $h = S_{max}/M$ the step sizes in time and asset, respectively. We will find approximations $\{\mathrm{p}_i^j\}$ of $\{\mathrm{p}(t^j, s_i)\}$ on the set of grid points $\{(t^j, s_i)\}$, where $t^j = jk$, $j = N, N-1, \cdots, 0$ and $s_i = ih$, $i = 0, 1, \cdots, M$.

First, to discretize the equation (4), we use the finite difference method based on the upwind scheme as following: for $j = N-1, \cdots, 0$,

$$\frac{\mathrm{p}_i^{j+1} - \mathrm{p}_i^j}{k} + \frac{1}{2}\sigma^2(t^j, s_i)s_i^2\frac{\mathrm{p}_{i-1}^j - 2\mathrm{p}_i^j + \mathrm{p}_{i+1}^j}{h^2} + (r-d)s_i D^\nu(\mathrm{p}_i^j) - r\mathrm{p}_i^j = F(t^j, s_i, \mathrm{q}(t^j)), \quad i = 1, 2, \cdots, M-1, \tag{6}$$

where

$$D^\nu(\mathrm{p}_i^j) = \begin{cases} \dfrac{\mathrm{p}_{i+1}^j - \mathrm{p}_i^j}{h}, & \text{if } r \geq d, \\ \dfrac{\mathrm{p}_i^j - \mathrm{p}_{i-1}^j}{h}, & \text{if } r < d. \end{cases}$$

Also, using the terminal and boundary conditions (5), we set

$$\mathrm{p}_i^N = \max(E - s_i, 0), \qquad i = 0,\ 1,\ \cdots,\ M, \tag{7}$$

$$\mathrm{p}_0^j = E, \qquad j = N,\ N-1,\ \cdots,\ 0, \tag{8}$$

$$\mathrm{p}_M^j = 0, \qquad j = N,\ N-1,\ \cdots,\ 0. \tag{9}$$

For each $j = N, \cdots, 0$, let p^j and b^j be the column vectors defined by

$$\mathrm{p}^j = (\mathrm{p}_0^j, \mathrm{p}_1^j, \cdots, \mathrm{p}_M^j)^{\mathrm{T}}, \quad b^j = (b_0^j, b_1^j, \cdots, b_M^j)^{\mathrm{T}},$$

where

$$b_0^j = b_M^j = 0, \quad b_i^j = kF(t^j, s_i, \mathrm{q}(t^j)), \quad i = 1, 2 \cdots, M-1.$$

Then, (6) - (9) form a set of linear system

$$A_j \mathrm{p}^j = \mathrm{p}^{j+1} + b^j, \quad j = N-1, N-2, \cdots, 0, \tag{10}$$

where, $A_j = (a^j_{mn})$ is the $(M+1)\times(M+1)$ tri-diagonal matrix with entries

$$
\begin{aligned}
&a^j_{0,0} = a^j_{M,M} = 1, \quad a^j_{0,1} = a^j_{M,M-1} = 0,\\
&a^j_{i,i-1} = k\left(-\frac{(\sigma^j_i)^2 i^2}{2} + \min(0, r-d)i\right), \quad i = 1, 2, \cdots, M-1,\\
&a^j_{i,i} = 1 + k\left((\sigma^j_i)^2 i^2 + |r-d|i + r\right), \quad i = 1, 2, \cdots, M-1,\\
&a^j_{i,i+1} = k\left(-\frac{(\sigma^j_i)^2 i^2}{2} - \max(0, r-d)i\right), \quad i = 1, 2, \cdots, M-1.
\end{aligned}
$$

Starting from p^N given by (7), we obtain the approximate solution by solving (10).

3 Parameter Estimation

Now we construct a finite dimensional approximation scheme for our optimization problem. First, given a positive integer L, let Q^L be the set of all functions in Q which is piecewise linear on the uniform mesh $\{\tau^\nu = \nu T/L\}_{0\le\nu\le L}$. For each $\nu = 0, \cdots, L$, let ϕ_ν denotes the standard hat function at τ^ν, i.e.,

$$
\phi_\nu(t) = \begin{cases} 1 + L(t-\tau^\nu)/T, & \tau^{\nu-1} \le t \le \tau^\nu,\\ 1 - L(t-\tau^\nu)/T, & \tau^\nu \le t \le \tau^{\nu+1},\\ 0, & \text{otherwise.}\end{cases}
$$

It is clear that the functions $\{\phi^\nu\}_{0\le\nu\le L}$ span Q^L, and hence the following problem is finite dimensional.

Problem $\mathrm{P}^{L,M,N}$. Find q^* that minimizes

$$
\mathrm{J}^{M,N}(\mathrm{q}) = \sum_{\nu=0}^{L} |\mathrm{p}^{M,N}(\tau^\nu, \mathrm{q}(\tau^\nu); \mathrm{q}) - E + \mathrm{q}(\tau^\nu)|^2 \tag{11}
$$

in Q^L, where $\mathrm{p}^{M,N}$ denotes the numerical solution described in the previous section.

We hope that each problem $\mathrm{P}^{L,M,N}$ has a solution $\mathrm{q}^{L,M,N}$ and that the sequence $\{\mathrm{q}^{L,M,N}\}$ converges to a solution of our original optimization problem. More specifically, if the following three conditions hold, then, it is called *function space parameter estimation convergent* [1]:

[H1] For each $L, M, N \in \mathbb{N}$, there exists a solution $\mathrm{q}^{L,M,N} \in \mathrm{Q}^L$ of Problem $\mathrm{P}^{L,M,N}$.

[H2] There exist increasing sequences $\{L_k\}$, $\{M_k\}$, and $\{N_k\}$ in $\mathbb{N}$ such that the resulting subsequence $\{\mathrm{q}^{L_k,M_k,N_k}\}$ converges to an element in Q.

[H3] Suppose that $\{L_k\}$, $\{M_k\}$, and $\{N_k\}$ are increasing sequences in $\mathbb{N}$ such that the corresponding subsequence $\{\mathrm{q}^{L_k,M_k,N_k}\}$ converges to $\mathrm{q}^* \in \mathrm{Q}$, then, q^* is a solution to the original problem.

There are a large number of choices in selecting an algorithm to minimize the cost functional (11). We adopted the Levenberg-Marquardt method [11], which has been most commonly used for the minimization problems with the least squares error functional as our problem. It is a type of quasi-Gauss-Newton method designed especially for least squares minimization, for which the Hessian of the cost functional can be approximated from the Jacobian of that.

To obtain an approximate solution of problem $\mathrm{P}^{L,M,N}$, we start with an initial guess $\mathrm{q}^{(0)} \in \mathrm{Q}^{L,M,N}$, and generate a sequence by $\mathrm{q}^{(k+1)} = \mathrm{q}^{(k)} + \delta^{(k)}$ until a suitable stopping criterion is satisfied. In each iteration we need to evaluate the Jacobian of the cost functional with respect to the parameter q, for which we use finite differences.

4 Numerical Results

To illustrate the function space parameter estimation convergence, we present two examples. The first one is the case of which the analytic solution is known, and the other corresponds to the classical constant-volatility model.

Example 1. Let $r = 0.09$, $d = 0.1$, $\sigma(t,s) = (1/s)\sqrt{2 \times 0.09}(-2e^t + 2e + 1)$, $E = 10$, and $T = 1$. Also, let G be a function such that

$$\mathrm{v}(t,s) = \begin{cases} (-2e^t + 2e + 1)e^{(2e^t+9-2e-s)/(-2e^t+2e+1)}, & s > \mathrm{s_p}(t), \\ 10 - s, & s \le \mathrm{s_p}(t), \end{cases} \tag{12}$$

and

$$\mathrm{s_p}(t) = 2e^t + 9 - 2e \tag{13}$$

become the solution of

$$\begin{aligned} &\frac{\partial \mathrm{v}}{\partial t} + 0.09(-2e^t + 2e + 1)^2 \frac{\partial^2 \mathrm{v}}{\partial s^2} - 0.01s\frac{\partial \mathrm{v}}{\partial s} - 0.09\mathrm{v} \\ &\quad = G(t,s) + [0.1s - 0.9 - G(t,s)]\, H(\mathrm{s_p}(t) - s), \end{aligned} \tag{14}$$

$$\mathrm{v}(t, \mathrm{s_p}(t)) = 10 - \mathrm{s_p}(t), \tag{15}$$

with the terminal and boundary conditions

$$\mathrm{v}(1,s) = \begin{cases} 10 - s, & s \le 9, \\ e^{9-s}, & s > 9, \end{cases} \tag{16}$$

$$\mathrm{v}(t,0) = 10, \tag{17}$$

$$\mathrm{v}(t,40) = (-2e^t + 2e + 1)e^{(2e^t+9-2e-40)/(-2e^t+2e+1)}. \tag{18}$$

In fact,

$$G(t,s) = \left[\frac{2e^t(2e^t + 9 - 2e - s)}{-2e^t + 2e + 1} + 0.01s\right] e^{(2e^t+9-2e-s)/(-2e^t+2e+1)}.$$

Table 1. Numerical results of Example 1

(L,M,N)	J(q)	$\|q^{L,M,N} - s_p\|_\infty$	$\|p^{M,N}(q^{L,M,N}) - v\|_\infty$
(2,80,32)	1.3516e-5	1.6106e-1	6.2971e-3
(4,160,64)	6.7679e-6	8.0148e-2	3.1787e-3
(8,320,128)	5.3258e-7	4.7070e-2	1.5964e-3
(16,640,256)	4.2231e-7	1.6655e-2	7.9994e-4

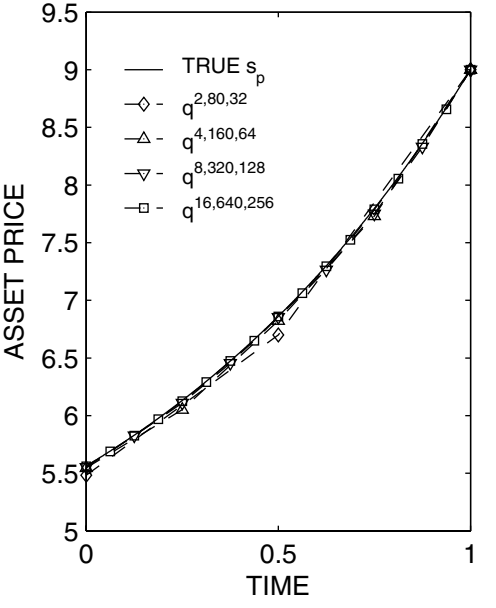


Fig. 1. Free boundary estimation in Example 1

Following the theory described in previous sections we took the domain $(0,1) \times (0,40)$ and consider the following parameter estimation problem: Find a function parameter q such that the solution $p(q) = p(t,s;q)$ of

$$\frac{\partial p}{\partial t} + 0.09(-2e^t + 2e + 1)^2 \frac{\partial^2 p}{\partial s^2} - 0.01s\frac{\partial p}{\partial s} - 0.09p = G(t,s) + [0.1s - 0.9 - G(t,s)]\, H(q(t) - s), \tag{19}$$

$$p(1,s) = e^{9-s} + (10 - s - e^{9-s})H(9-s), \quad p(t,0) = 10, \quad p(t,40) = 0 \tag{20}$$

satisfies the condition

$$p(t, q(t); q) = 10 - q(t), \quad 0 \le t \le 1. \tag{21}$$

It is well known that the value of s_p at maturity date is given by $\min(E, (r/d)E)$, see, for example, [5,14]. So, we took $q(1) = 9$ in the estimation process.

Table 2. The sequence of cost values in Example 2

(L,M,N)	$J(q^{(0)})$	$J(q^{(1)})$	$J(q^{(2)})$	$J(q^{(3)})$	$J(q^{(4)})$	$J(q^{(5)})$
(5,100,50)	4.41e+0	2.88e-1	3.55e-2	3.87e-3	9.82e-4	2.40e-4
(10,200,100)	7.87e+0	4.22e-1	3.46e-2	2.65e-3	3.37e-4	1.55e-4
(25,400,200)	1.83e+1	8.01e-1	5.01e-2	1.89e-3	1.15e-4	1.74e-5
(50,800,400)	3.57e+1	1.58e+0	8.33e-2	1.86e-3	8.39e-5	9.06e-6
(100,1600,800)	7.04e+1	3.27e+0	1.62e-1	2.41e-3	5.21e-5	6.08e-6
(200,3200,1600)	1.40e+2	5.87e+0	2.69e-1	4.26e-3	4.31e-5	1.45e-6

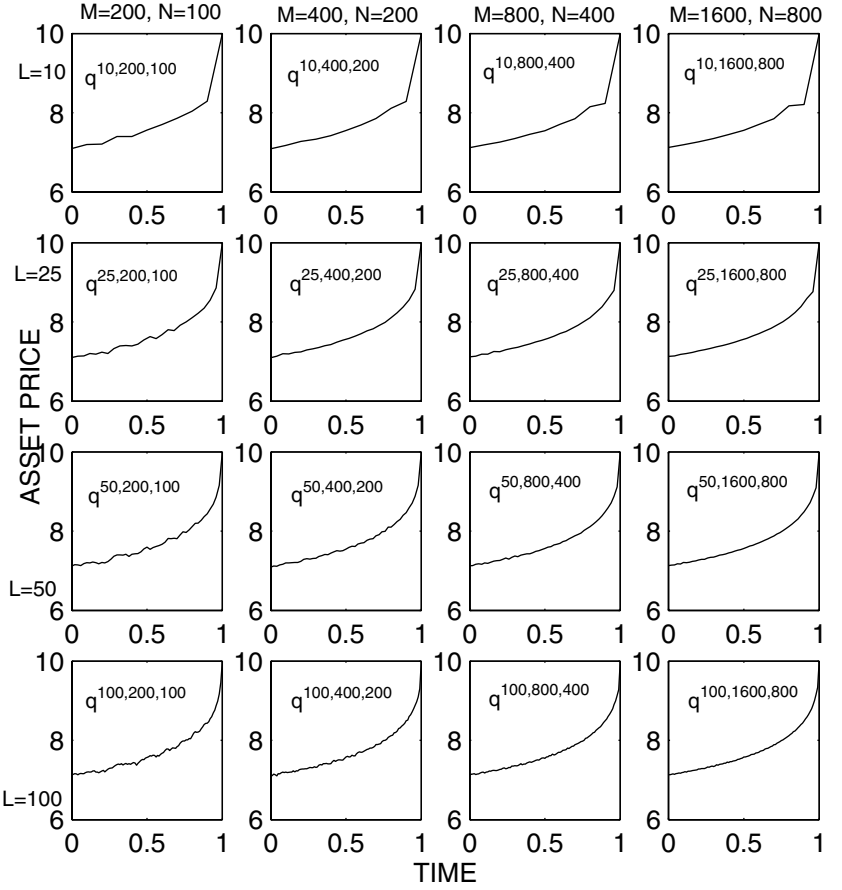


Fig. 2. Optimal exercise curve estimation in Example 2

Table 1 and Figure 1 shows the expected convergence. Here, $q^{L,M,N}$ is the estimated approximation of the true free boundary s_p, and $p^{M,N}(q^{L,M,N})$ is the corresponding approximation of the true option value v. As initial guesses

Table 3. Numerical current prices of the American put option in Example 2

(L,M,N) \ s	7.0	8.5	10.0	11.5	13.0
(5,100,50)	2.9904	1.7464	0.9581	0.5036	0.2558
(10,200,100)	2.9995	1.7425	0.9533	0.4980	0.2508
(25,400,200)	3.0000	1.7390	0.9504	0.4948	0.2487
(50,800,400)	3.0000	1.7381	0.9490	0.4935	0.2477
(100,1600,800)	2.9998	1.7374	0.9483	0.4929	0.2472
(200,3200,1600)	2.9999	1.7372	0.9480	0.4926	0.2469
BTM	3.0000	1.7371	0.9475	0.4925	0.2467

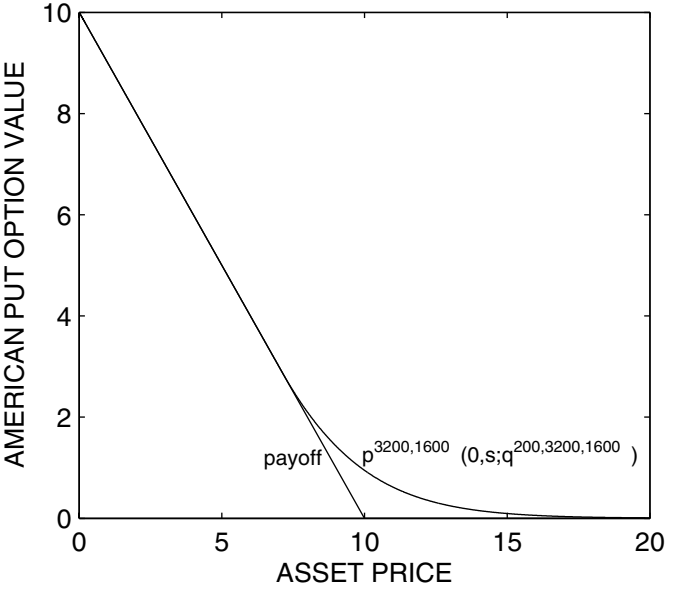


Fig. 3. Numerical current prices of the American put option in Example 2

in using Levenberg-Marquardt algorithm we took the constant function 9. The maximum errors $\|\mathrm{q}^{L,M,N} - \mathrm{s_p}\|_\infty$ and $\|\mathrm{p}^{M,N}(\mathrm{q}^{L,M,N}) - \mathrm{v}\|_\infty$ means the quantity

$$\max_{0\le\nu\le L} |\mathrm{q}^{L,M,N}(\tau^\nu) - \mathrm{s_p}(\tau^\nu)| \text{ and } \max_{0\le i\le M,\, 0\le j\le N} |\mathrm{p}^{M,N}(t^j, s_i; \mathrm{q}^{L,M,N}) - \mathrm{v}(t^j, s_i)|,$$

respectively, where $\tau^\nu = \nu/L$, $t^j = j/N$, and $s_i = 40i/M$.

Example 2. In this example, we test our approximation scheme with the classical American put option, in which $r = 0.07$, $d = 0.01$, $\sigma = 0.3$, $E = 10$, and $T = 1$. The forcing term is $F(t, s, \mathrm{s_p}) = (ds - rE)H(\mathrm{s_p}(t) - s)$. We take $S_{max} = 40$ and proceed as in Example 1.

Table 2 reports the sequence of the cost values, and Figure 2 the estimated parameters, from which we see the expected convergence result.

Table 3 and Figure 3 show the corresponding current prices of American put options, from which we see that our method give almost the same result as the one obtain by BTM (Binomial Tree Method)[2,3], which is most popularly used method for American option pricing. In applying BTM we use 1000 depth of the time steps.

5 Conclusion

In this paper, we have dealt with the free boundary of American put option which has smile volatility depending on time variable t and asset variable s. We propose a parameter estimation technique as a numerical scheme to compute the pair of put option value and the free boundary. Numerical results shows that our approximation scheme enjoys the function space parameter estimation convergence. This method produces almost the same values of the put option price as the binomial tree method does.

References

1. Banks, H. T., Kunisch, K.: Estimation Techniques for Distributed Parameter Systems. Birkhäuser, Boston (1989)
2. Cox, J.C., Ross, S.A., Rubinstein, M.: Option pricing: A simplified approach. Journal of Financial Economics **7** (1979) 229–264
3. Hull, J.C.: Introduction to Futures and Options Markets. 3rd edn. Prentice-Hall, New Jersey (1998)
4. Jamshidian, F.: An analysis of American options. Review of Futures Markets **11** (1992) 72–80
5. Kholodnyi, V.A.: A nonlinear partial differential equation for American options in the entire domain of the state variable. Nonlinear Analysis, TMA **30** (1997) 5059–5070
6. Kuske, R.A., Keller, J.B.: Optimal exercise boundary for an American put option. Appl. Math. Finance **5** (1998) 107–116
7. Lindberg, P., Marcusson, G., Nordman, G.: Numerical analysis of the optimal exercise boundary of an American put option, Internal Report **2002:7**, Uppsala University (2002)
8. Mckean, H.P., Jr.: Appendix: a free boundary problem for the heat equation arising from a problem in mathematical economics. Industrial Management Review **6** (1965) 32–39
9. Merton, R.C.: Theory of rational option pricing. Bell Journal of Economics and Management Science **4** (1973) 141–183
10. Roos, H.G., Stynes, M., Tobiska, L.: Numerical Methods for Singularly Perturbed Differential Equations. Springer-Verlag, Berlin (1996)
11. Press, W.H., Teukolsky, S.A., Vettering, W.T., Flannery, B.P.: Numerical Recipes in C. Cambridge University Press (1992)
12. Ševčoviç, D.: Analysis of the free boundary for the pricing of an American call option. European J, Appl. Math. **12** (2001) 25–37
13. Shu, J., Gu, Y., Zheng, W.: A novel numerical approach of computing American option. Int. J. Found. Comut. Sci. **13** (2002) 685–693
14. Wilmott, P., Dewynne, J., Howision, S.: Option Pricing: Mathematical Models and Computation. Oxford Financial Press. Oxford (1995)

Forecasting the Volatility of Stock Index Returns: A Stochastic Neural Network Approach

Chokri Slim

Université de la Manouba
Institut Supérieur de Comptabilité
et d'Administration des Entreprises, Laboratoire BESTMOD
41, Rue de la Liberté, Cité Bouchoucha, 2000 Le Bardo,Tunisia
chokri.slim@iscae.rnu.tn

Abstract. In this paper we are concerned with the volatility modelling of financial data returns, especially with the nonlinear aspects of these models. Our benchmark model for financial data returns is the classical GARCH(1,1) model with conditional normal distribution. As a tool for its nonlinear generalization we propose a Stochastic neural network (SNN) to the modelling and forecasting the time varying conditional volatility of the TUNINDEX (Tunisia Stock Index) returns. Such specification also helps to investigate the degree of nonlinearity in financial data controlled by the neural network architecture. Our empirical analysis shows that out-of-simple volatility forecasts of the SNN are superior to forecasts of traditional linear methods (GARCH) and also better than merely assuming a conditional Gaussian distribution.

1 Introduction

Forecasts of financial market volatility play a crucial role in financial decision making and the need for accurate forecasts is apparent in a number of areas, such as option pricing, hedging strategies, portfolio allocation and Value-at-Risk calculations. Due to its critical role the topic of volatility forecasting has however received much attention and the resulting literature is considerable. A comprehensive survey of the findings in the volatility forecasting performance literature is given in Poon and Granger [1].

One of the main sources of volatility forecasts are historical parametric volatility models such as Autoregressive Conditional Heteroskedasticity (ARCH) and generalized (GARCH), ([2]and [3]) which assume a conditional normal distribution and the Stochastic Volatility (SV) models [4]. The parameters in these models are estimated with historical data and subsequently used to construct out-of-sample volatility forecasts. Studies comparing the forecasting abilities of the various volatility models have been undertaken for a number of stock indices and the general consensus appears to be that those models that attribute more weight to recent observations outperform others ([5] and [6]).

In the meantime there exists a large body of literature dealing with extensions of the GARCH approach. Recently neural networks have been proposed to

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 935–944, 2004.
© Springer-Verlag Berlin Heidelberg 2004

model leverage effects in a nonlinear and semi-nonparametric way [7]. The second direction in which GARCH models have been generalized, is the specification of the conditional density [8].

In this paper, we present a Stochastic Neural Network (SNN), which approximates the conditional density of the time varying variance, and they allow for heteroskedastic dependence in the TUNINDEX (Tunisia Stock Index) returns. The main idea is to let the SNN predict the parameters of the conditional density in dependence of the data. Besides that, The SNN based model allows for nonlinear dependences in the conditional mean and in the conditional variance.

Accordingly, the rationale for this paper is to investigate the predictive power of alternative forecasting models of volatility, both from a statistical and an economic point of view. We examine the use of our SNN, and some GARCH models for forecasting volatility, with an application to the TUNINDEX returns.

The organization of this paper is as follows. Section 2 briefly summarizes the candidate models of the stock returns, including finite-variance family and infnite-variance family. In section 3 the concept of neural networks is presented and extended to the SNN way to allows for GARCH effects. The empirical analysis of the volatility models with special emphasis on the prediction performance is described in detail in section 4. Section 5 concludes and gives suggestions for further research.

2 Volatility Models

There are three general classes of volatility models in widespread use. The first type formulates the conditional variance directly as a function of observable. The simplest examples here are the ARCH and GARCH models which will be discussed in some detail in this section. The second general class formulates models of volatility that are not functions purely of observable. These might be called latent volatility or (misleadingly) stochastic volatility models. the third classes is the neural network models that its efficiently approximates a wide class of nonlinear relations. In the context of heteroskedastic time series modeling, neural networks have been used to capture volatility models [7]. The authors encouraged researchers to " go beyond linearity ", as well as to investigate in more detail the economic rationale " underlying the " good performance of nonlinear models.

2.1 Stylized Facts

A number of stylized facts about the volatility of financial asset prices have emerged over the years, and been confirmed in numerous studies. A good volatility model, then, must be able to capture and reflect these stylized facts:

- The clustering of large moves and small moves (of either sign) in the price process was one of the first documented features of the volatility process of asset prices. The large changes in the price of an asset are often followed by other large changes, and small changes are often followed by small changes.

The implication of such volatility clustering is that volatility shocks today will influence the expectation of volatility many periods in the future.
- Volatility clustering implies that volatility comes and goes. Thus a period of high volatility will eventually give way to more normal volatility and similarly, a period of low volatility will be followed by a rise. Mean reversion in volatility is generally interpreted as meaning that there is a normal level of volatility to which volatility will eventually return. More precisely, mean reversion in volatility implies that current information has no effect on the long run forecast.
- Many proposed volatility models impose the assumption that the conditional volatility of the asset is affected symmetrically by positive and negative innovations. For equity returns it is particularly unlikely that positive and negative shocks have the same impact on the volatility. This asymmetry is sometimes ascribed to a leverage effect and sometimes to a risk premium effect.
- Most of the volatility characteristics outlined above have been univariate, relating the volatility of the series to only information contained in that series. Of course, no one believes that financial asset prices evolve independently of the market around them, and so we expect that other variables may contain relevant information for the volatility of a series.
- It is well established that the unconditional distribution of asset returns has heavy tails. Typical kurtosis estimates range from 4 to 50 indicating very extreme non normality. This is a feature that should be incorporated in any volatility model.

2.2 ARCH and GARCH Models

The autoregressive conditional heteroskedasticity (ARCH) models introduced by Engle [2] and their generalization,the so called GARCH models [3] have been the most commonly employed class of time series models in the recent finance literature.

The core of the ARCH models is that the conditional distribution of the next return is a normal distribution with a variance depending in a linear fashion on a number of previous errors. The conditional mean, is usually specified as a constant or as a linear function of the last return:

$$f(r_t|x_t) = N(\mu_t, \sigma_t^2) \tag{1}$$

$$\mu_t = a + br_{t-1} \tag{2}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 \tag{3}$$

$$\varepsilon_t = r_t - \mu_t \tag{4}$$

where x_t denotes the information available at time t, and $N(\mu_t, \sigma_t^2)$ is the density of normal distribution of mean μ_t and variance σ_t^2, ε_t is the prediction errors at time t.

The conditional variance σ_t^2 is the volatility of return series at time t estimated by ARCH(q) model.

For the ARCH(q) model, in most empirical studies, q has to be large. This motivates Bollerslev [3] to use the GARCH(p, q) specification which is defined as:

$$f(r_t|x_t) = N(\mu_t, \sigma_t^2) \tag{5}$$

$$\mu_t = a + br_{t-1} \tag{6}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \tag{7}$$

$$\varepsilon_t = r_t - \mu_t \tag{8}$$

The number of publications dealing with extensions of the GARCH model is tremendous. For a comprehensive overview the reader is referred to [1].

2.3 Neural Networks Models

Artificial neural networks(ANN) are a class of nonlinear regression models, that can flexibly be used as parametric, semi- and non-parametric methods. Their primary advantage over more conventional techniques lies in their ability to model complex, possibly nonlinear processes without assuming prior knowledge about the data-generating process. This makes them particularly suited for financial applications. In this area such methods showed much success and are therefore often used, see Chokri et al ([9] and [10]). A complete description of a Neural Networks theory and the application of neural networks to the problem of nonlinear system identification and prediction can be found in Hertz [11]. Figure 1 show a neural network architecture with three layers (input, hidden and output).

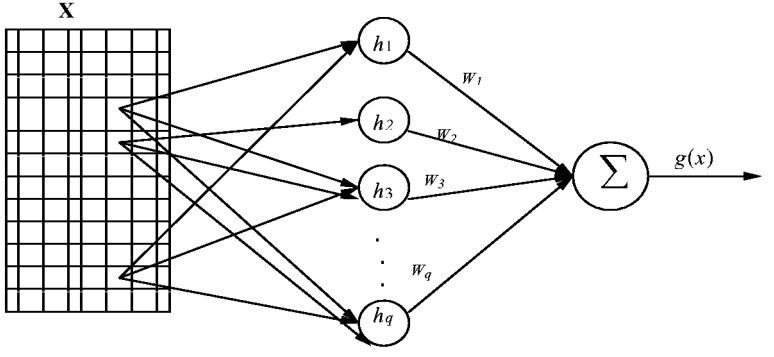


Fig. 1. A neural network architecture

A continuous function $g(x)$ can be approximated by the net of figure.1 as a linear combination:

$$g(x) \approx \sum_{i=1}^{q} w_i h_i(x)$$

where w_i is the strength of connections between hidden neurons and the output response, q is the number of hidden neurons and h_i is the activation function.

3 The Stochastic Neural Net

There have been a number of methods on using neural networks for density estimation. The majority of the approaches tend to be parametric in nature, and therefore share many of the limitations of the parametric approach [8].

3.1 Estimation Technique

We suppose that the conditional density of the return series is unknown with conditional mean μ_t and conditional variance σ_t^2. The conditional density can be modeled from a mixing density:

$$f(r_t|x_t) = \sum_{k=1}^{l} \beta_{k,t} f(r_t|x_{k,t}) \tag{9}$$

where $\beta_{k,t}$ can be regarded as prior probability (conditioned on x), $f(r_t|x_{k,t})$ are the component densities, generally a Gaussian densities. We consider in this section, a Stochastic Neural Networks (SNN) to identify the model in (9). The prior probability and the conditional mean and the conditional variance of the unknown conditional density can be approximated by a sub-neural networks as in figure 1, then:

$$\beta_{k,t} = \sum_{i=1}^{q_1} w_i^{(\beta_k)} h_i(x_t) \tag{10}$$

$$\hat{\mu}_{k,t} = \sum_{j=1}^{q_2} w_j^{(\mu_k)} h_j(x_t) \tag{11}$$

$$\hat{\sigma}_{k,t}^2 = \left(\sum_{m=1}^{q_3} w_m^{(\sigma_k)} h_m(x_t) \right)^2 \tag{12}$$

where q_1, q_2 and q_3 are the number of hidden nodes in each sub-networks and $h(x)$ is the hyperbolic tangent activation function.

In order to satisfy the constraint $\sum_{k=1}^{l} \beta_{k,t} = 1$ and $\beta_{k,t} > 0$ The actual value of $\beta_{k,t}$ is calculated as:

$$\beta_{k,t} = \frac{\exp(\sum_{i=1}^{q_1} w_i^{(\beta_k)} h_i(x_t))}{\sum_{k=1}^{l} \exp(\sum_{i=1}^{q_1} w_i^{(\beta_k)} h_i(x_t))} \tag{13}$$

The overall estimate model (9) can be represented in figure 2, which is a two layered structure.

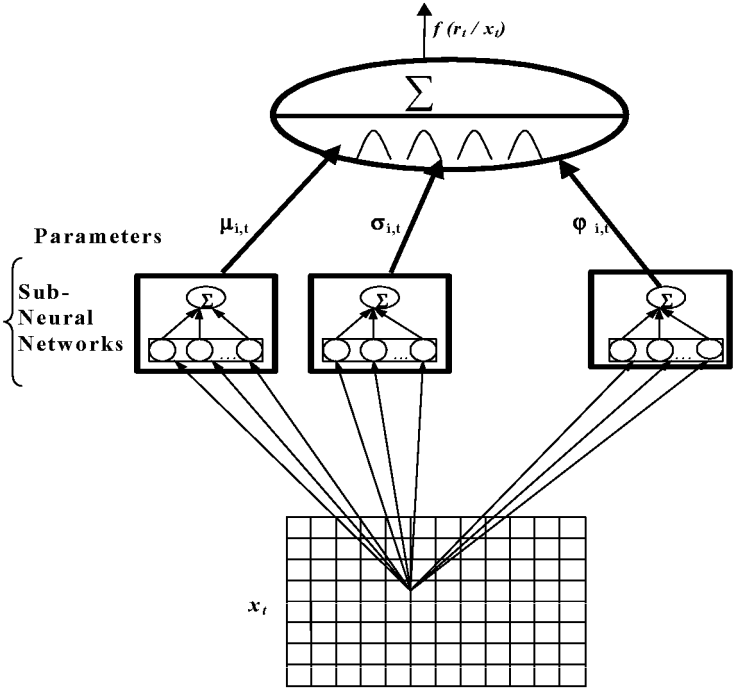


Fig. 2. The stocahstic Neural Networks

In order to estimate the parameters of each sub-networks, We define the likelihood function (assuming that the training data is being drawn independently from the mixture distributions):

$$\mathrm{L} = \prod_{t=1}^{n} f(r_t|x_t) \tag{14}$$

We use an error function to be the negative logarithm of the likelihood function:

$$E = -\ln \mathrm{L} \tag{15}$$

In order to minimize E we apply an advanced gradient descent algorithm which uses information about of the error function with respect to the parameters of the mixture model and each sub-networks. Using the chain rule, for each parameter the gradient according to the error-function E can be calculated and the parameters be adjusted accordingly. This is possible since we chose the functions to be differentiable with respect to each parameter [8]. E can be minimized with many numerical-search algorithms. Here we use a gradient backpropagation algorithm (Chokri et al [9] and [10]).

4 Empirical Analysis

In this section we first describe the data sets which are used to analyze the performance of the volatility models. Then some error measures which qualify various aspects of the forecasting performance of the models, are specified. After reporting the in-sample results, special attention is given to the out-of-sample performance of the various models.

4.1 Summary of the Data

We use daily close price data on the Tunisia index (TUNINDEX), over the period Jan 05, 1998 to Dec. 31, 2002, representing 1153 observations. From the series of TUNINDEX closing values S_t, the corresponding series of returns r_t is calculated:

$$r_t = 100 \log \left(\frac{S_t}{S_{t-1}} \right)$$

Figures 3 plot the price level and the returns on the index over the sample period.

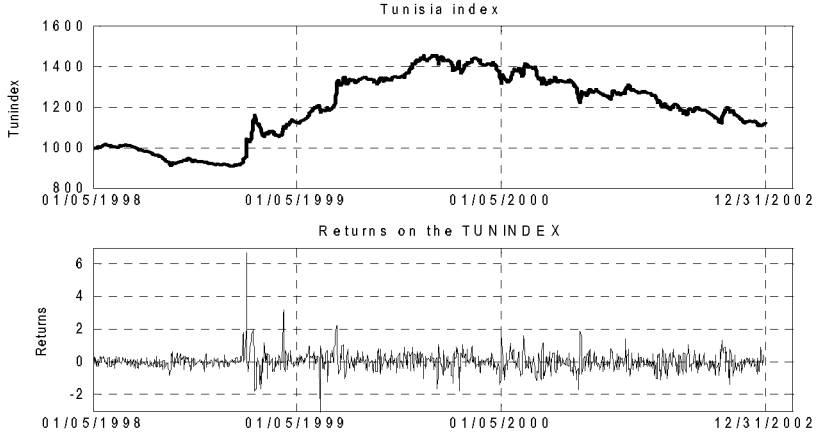


Fig. 3. Price level and returns on the TUNINDEX over the sample period

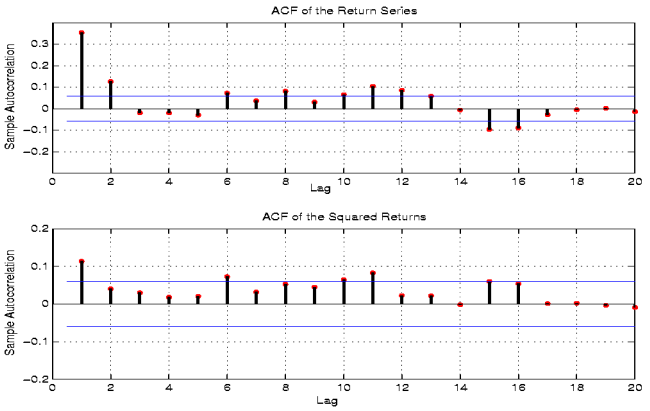


Fig. 4. Price level and returns on the TUNINDEX over the sample period

An analysis of the correlogram of the returns, presented in figure 4, indicates dependence in the mean of the series. The correlogram of the squared returns, indicates substantial dependence in the volatility of returns. Using formal hypothesis tests such as the Ljung-Box-Pierce Q-test and Engle's Arch test [2] to quantify the proceeding qualitative checks for correlation, these tests shows significant evidence in support of GARCH effects (i.e., heteroscedasticity).

4.2 Application and Results

The data has been divided into two sets, an in-sample part with 600 and an out-of-sample part with 552 returns. Now three models were estimated, GARCH(1,1) models as a benchmark, EGARCH(1,1), and a neural networks SNN (1,5,2), with one input, five hidden nodes and two mixtures gaussians density.

Forecast Evaluation: The natural error measure for all models is the loss function E in (15). In addition to the loss function two alternative error measures are used the normalized mean absolute error NMAE relates the mean absolute error MAE of the volatility model of the naive model ($\hat{\sigma}_t^2 = r_t^2$) and the hit rate HR is the relative frequency of correctly predicted increases and decreases of volatility.

$$\text{NMAE} = \frac{\sum_{t=1}^{N} \left| r_t^2 - \hat{\sigma}_t^2 \right|}{\sum_{t=1}^{N} \left| r_t^2 - r_{t-1}^2 \right|} \tag{16}$$

$$\text{HR} = \frac{\sum_{t=1}^{N} \gamma_t}{N} \quad \text{with} \quad \gamma_t = \begin{cases} 1 : (r_t^2 - \hat{\sigma}_t^2)(r_t^2 - r_{t-1}^2) \\ 0 : \qquad\qquad \text{else} \end{cases} \tag{17}$$

In-sample Results: A GARCH(1,1), EGARCH(1,1), and a neural networks SNN (1,5,2) were fitted to the in-sample set by minimizing the loss function E.

Table 1. In-sample statistics for the GARCH(1,1), EGARCH(1,1) and SNN(1,5,2)

Model	E	NMAE	HR
GARCH(1,1)	1.46	0.82	0.79
EGARCH(1,1)	1.33	0.72	0.81
SNN(1,5,2)	**1.22**	**0.69**	**0.68**

Table 2. Out-of-sample statistics for the GARCH(1,1), EGARCH(1,1) and SNN(1,5,2)

Model	E	NMAE	HR
GARCH(1,1)	1.32	0.69	0.66
EGARCH(1,1)	1.22	0.72	0.69
SNN(1,5,2)	**1.16**	**0.67**	**0.64**

Table 1 summarizes the performance of the models on the in-sample set. The SNN(1,5,2) model achieves the lowest value of the all performance errors.

As one can see from the in-sample results in Table 1, the performance of the models heavily depends on the errors measure. The SNN(1,5,2) for instance, is better than the two other models with respect to all performance errors. The EGARCH (1,1) model is better than the GARCH(1,1) model in term of the all errors.

Our main interest lies in the out-of-sample performance of the models where the number of parameters does not have to be taken into account when comparing models.

Out-of-sample Results: It has been emphasized in the literature that a volatility model should always be tested ou-of-sample. Table 2 summarizes the performance of the models with respect to the usual error measures. As in the in-sample analysis, the performance is closely related to the inspected error measure. The SNN(1,5,2) model exhibit the smallest measure errors and the GARCH(1,1) is better than the EGARCH(1,1) with respect to the NMAE and HR.

5 Summary and Conclusions

In this paper we study a general class of asset return models that nests several existing models as special cases. In particular we specify a Stochastic Neural Network and analyze the impact of nonlinearity and non Gaussian behavior on the predictive power of conditional variances. We use the TUNINDEX returns series over the sample period Jan 1998 to December 2002 to evaluate the in-sample and out-of-sample forecasting abilities of three different specifications: (i) a simple GARCH(1,1) process with conditional normal distributions, (ii) a EGARCH(1,1) model, (iii) SNN(1,5,2) with two mixture gaussians density. After comparing the forecasting performance of three models, we find that the SNN model is superior according to the performance errors.

Still further research has to be conducted on the proposed framework like investigating, how more complex mixture models perform on these tasks as well as how these models perform, if the whole density forecast is considered. Of further interest is selecting the right architecture of neural networks is still to be investigated in the area of statistical selection theory. All the models examined in this paper belong to the univariate time series family. The multivariate times series is an interesting extension to our future work.

References

1. Poon, S.H., Clive, W. J.G.: Forecasting Volatility in Financial Markets: A Review. Journal of Economic Literature. Vol. XLI (2003) 478–539
2. Engle, R.: Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation. Econometrica.50 (1982) 987–1008
3. Bollerslev, T.: Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics. 31 (1986) 307–327
4. Andersen, T., Bollerslev, T., Diebold, F.X. and Labys, P.: The distribution of realized exchange rate volatility. Journal of the American Statistical Association 96 (2001) 42–55
5. Shephard, N.: Statistical aspects of ARCH and stochastic volatility. In Cox, D. R., Hinkley, D. V., Time Series Models in Econometrics. Finance and Other Fields Barndorff-Nielson. O. E. (eds), London: Chapman & Hall. (1996) 1–67
6. Hentschel, L.: All in the family: Nesting symmetric and asymmetric GARCH models. Journal Financial Economics. 39 (1995) 71–104
7. Donaldson, R.G. and Kamstra, M.: An Artificial Neural Network - GARCH Model for International Stock Return Volatility. Journal of Empirical Finance. 4 (1997) 17–46
8. Bishop,W.: Mixture Density Networks. Technical Report NCRG/94/004. Neural Computing Research Group. Aston University. Birmingham. (1994)
9. Chokri, S., Abdelwahed, T.: Neural Network for Modeling Financial Time Series: A New Approach. V.Kumer et al. (EDS). Springer-Verlag Berlin LNCS. 2669 (2003) 236–245
10. Chokri, S., Abdelwahed, T.: Neural Network for Modeling Nonlinear Time Series: A New Approach. Springer-Verlag Berlin LNCS. 2659 (2003) 159–168
11. Hertz, J., Krogh, A., and Palmer, R.:Introduction to the Theory of Neural Computation. Santa Fe Institute Studies in the Science of Complexity. Amesterdam: Addison-Wesley. (1991)

A New IP Paging Protocol for Hierarchical Mobile IPv6*

Myung-Kyu Yi and Chong-Sun Hwang

Dept. of Computer Science & Engineering Korea University,
1,5-Ga, Anam-Dong, SungBuk-Gu, Seoul 136-701, South Korea
{kainos, hwang}@disys.korea.ac.kr

Abstract. In contrast to the advantages of Mobile IP, updating the location of an mobile node incurs high signaling cost if it moves frequently. Thus, IP paging protocols for Mobile IP (P-MIP) have been proposed to avoid unnecessary registration signaling overhead when a mobile node is idle. However, it requires additional paging cost and delays associated with message delivery when a correspondent node sends packets to the idle mobile node. These delays greatly influence the quality of service (QoS) for multimedia services. Also, P-MIP does not consider a hierarchical mobility management scheme, which can be reduce signaling cost by the significant geographic locality in user mobility pattern. Thus, we propose a novel IP paging protocol which can be used in hierarchical Mobile IPv6 architecture. In addition, our proposal can reduce signaling cost for paging and delays using the concept of the *anchor-cell*. The cost analysis presented in this paper shows that our proposal has superior performance when the session-to-mobility ratio value of the mobile node is low.

1 Introduction

Mobile IP provides an efficient and scalable mechanism for host mobility within the Internet. Using Mobile IP, nodes may change their point of attachment to the Internet without changing their IP address. However, Mobile IPv6 (MIPv6) results in a high signaling cost to update the location of an Mobile Node (MN) if it moves frequently[1]. Thus, IP paging protocols for Mobile IP (P-MIP) approaches have been proposed to reduce signaling load in the core Internet and power consumption of MNs[2,3]. However, it requires additional signaling cost for paging and incurs high paging delays associated with message delivery if the Correspondent Node (CN) sends packets to the idle MN. These delays greatly influence the quality of service (QoS) for multimedia services. In addition, P-MIP does not consider a hierarchical mobility management scheme which aims to reduce the signaling load due to user mobility. In this paper, we propose a novel IP paging protocol (PHMIPv6) which can be used in hierarchical Mobile

* This work was supported by grant No. R01-2002-000-00235-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 945–954, 2004.
© Springer-Verlag Berlin Heidelberg 2004

IPv6 architecture. Also, we will employ the *anchor-cell* concept and techniques used in reducing signaling cost for paging and delays[4]. In our proposal, each MN has the anchor-cell where the MN stay in a cell for a long time. Whenever the idle MN enters or leaves its anchor-cell, it sends a Binding Update (BU) message to the Mobility Anchor Point (MAP) even though the MN is in idle mode. With a little increase in signaling cost for location updating, the signaling cost for paging and paging delays decreases dramatically. The rest of the paper is organized as follows. Section 2 illustrates the system model used in our proposal. Section 3 describes the proposed protocol of location update and packet delivery using the concept of the anchor-cell. Section 4 shows the evaluation of proposed system's performance and analysis of results. Finally, conclusions are presented in Section 5.

2 System Description

First of all, we define *MAP domain* as the highest level of our mobile network as shown in Fig. 1. As with most existing Hierarchical MIPv6 (HMIPv6)[5], we use an MAP in order to deploy two levels of hierarchies. The MAP domain can be organized into one or several paging areas, assuming that the MAP is in charge of the paging process for the paging area. In our proposal, we assume that each base station acts as an Access Router (AR) for the MN and the paging area is assumed to be preconfigured. The IP-based core network operates independently of the radio access network technology. The AR interacts with the radio access network consisting of base station and access point covering a specific geographic area. We assume that each AR always knows the exact location of an MN so that it can forward all packets destined to the idle MN. Each paging area is identified as having an unique Paging Area Identifier (PAI) and consists of two or more networks which are identified by the network prefix as part of their Internet address. Since the MAP has a paging table for the network prefixes and PAIs, it can get an MN's PAI from the MN's on-link Care-of Address (LCoA). In our proposal, we will employ the anchor-cell concept and techniques used in minimizing signaling cost for paging and delays as discussed in [4]. The anchor-

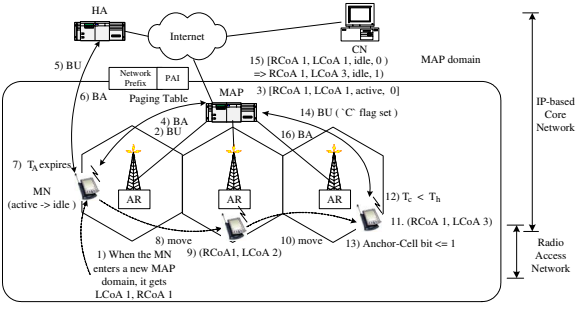


Fig. 1. The System Architecture of Our Proposed Protocol

cell can be defined as the cell that the MN stays in longer than the time period. The basic idea is based on the Stop-or-Move Mobility (SMM) model[6]. Based on the SMM model, we propose a new IP paging protocol which does not incur paging cost and delays when an idle MN stays in the anchor-cell. Each MN has a value of T_c and T_h. While T_c is the cell residence time that an MN stays in a cell, T_h is the value of the time threshold to decide whether the idle MN is in the anchor-cell or not. If the idle MN enters a new cell, it sets the cell residence time T_c to zero. If T_c is larger than T_h, the current cell that the MN is residing becomes its new anchor-cell. Each MN is in one of two operational modes, active or idle. In the same way as [3], these modes are governed by the active timer T_A in both the MN and serving MAP. Each MN and MAP has a value of the anchor-cell bit. The initial value of the anchor-cell bit is zero. If an idle MN stays in the anchor-cell, the anchor-cell bit is set to 1. In this case, the MAP sets the value of the anchor-cell bit to 1 in its binding cache for the MN by receiving the BU message from the idle MN. To support this operation, we propose to extend the HMIPv6 BU message with an extra flag 'anChor' (C) taken from the reserved field. If the 'C' flag is set, it indicates that the MN is staying in the anchor-cell.

3 Protocol Description

This section describes the location update and packet delivery procedure.

3.1 Location Update Procedure

Each MN has two addresses, an RCoA on the MAP's link and an on-link CoA (LCoA). The MN registers with its MAP and Home Agent (HA) by sending a BU message. When an MN is in active mode, it operates in exactly the same manner as in the existing HMIPv6[5]. However, when the active timer expires, the MN enters idle mode. The idle MN registers with the MAP only under the following conditions:

- An idle MN detects that it has moved to a new paging area or MAP domain.
- An idle MN's binding cache lifetime expires.
- An idle MN receives the paging request message from the AR.
- An idle MN's cell residence time T_c is larger than T_h.
- An idle MN leaves the anchor-cell.

Fig. 1 presents a simple location update scenario for location update by sending the BU message.

Case 1. If an idle MN enters a different MAP domain,

1) it sets the value of the anchor-cell bit to zero and registers with the HA and MAP by sending a BU messages with 'C' flag unset.
2) Then, the MAP sets a value of the anchor-cell bit in its binding cache for the MN to zero.

3) Finally, the MAP sends a BA message to the MN.

Otherwise, if an idle MN moves within the same MAP domain, the following cases take place:

Case 2-1. When an idle MN moves a different paging area:

1) It receives the new PAI from the Router Advertisements message.
2) Thus, it detects that it moved into a new paging area.
3) The idle MN must sends a BU message with 'C' flag unset to the MAP.
4) Finally, the MAP sets the value of the anchor-cell bit to zero in its binding cache for the MN.

Case 2-2. Otherwise, when an idle MN moves a same paging area,

1) the idle MN sets the value of cell residence timer T_c to zero.
2) If T_c is larger than T_h before leaving the cell, the idle MN sets a value of the anchor-cell bit to 1.
3) Then, it sends a BU message to the MAP with 'C' flag set.
4) After receiving the BU message from the idle MN, the MAP updates its binding cache entry for the MN and changes the anchor-cell bit to 1.
5) Finally, the MAP sends a BA message to the MN.
6) Whenever the idle MN moves out of the anchor-cell after previously setting the value of the anchor-cell bit to 1, it changes the value of the anchor-cell bit to zero and sends a BU message to the MAP with 'C' flag unset.
7) After receiving the BU message with 'C' flag unset, the MAP updates its binding cache entry for the MN and changes the anchor-cell bit to zero.

3.2 Packet Delivery Procedure

When a CN sends packets to the idle MN, it first sends packets to the MAP. Then, the MAP checks the MN's operational mode in its binding cache for the MN. If the MN is in an active mode, the MAP sends packets to the MN using the MN's LCoA. However, if the MN is in an idle mode, it checks the value of the anchor-cell bit in its binding cache for the MN. If the value of the anchor-cell bit is equal to 1, then the MAP sends packets to the MN using the LCoA. Finally, the MAP changes the operational mode to active. Otherwise, if the value of the anchor-cell bit is equal to zero, then the MAP immediately begins to buffer all packets destined to the MN. The MAP sends the paging request messages to the other ARs that have the same number of PAI. The AR broadcasts the paging request message to the MN. When the MN receives a paging request message, it sends a BU message to the MAP with 'C' flag unset. When the MAP receives a BU message from the idle MN, it sends a BA message to be piggybacked onto the buffered packets. Finally, the MAP changes the operational mode in its binding cache for the MN to active mode and sets the value of the anchor-cell bit to zero.

4 Performance Analysis

In this section we provide the mathematical analysis needed to evaluate the total signaling cost per unit time. We will compare our proposed scheme called PHMIPv6 with HMIPv6[5].

4.1 Location Update Cost in HMIPv6

The performance metrics is the total signaling cost which is the sum of the location update cost and packet delivery cost. For simplicity, we assume that an MAP domain is assigned to only one paging area. Thus, all ARs have the same PAI within a MAP domain. We define the costs and parameters used for the performance evaluation of location update as follows[7]:

- a_h: The location update processing cost at the HA.
- a_m: The location update processing cost at the MAP.
- l_{hm}: The average distance between the HA and the MAP.
- l_{mn}: The average distance between the MAP and the MN.
- δ_U: The proportionality constant for the location update.
- m: The number of times that the MN changes its point of attachment.
- M: The random variable before each MN moves out a domain.
- κ: The total number of cells within a domain.
- N: The total number of cells.
- T_c: The average residence time of an MN in each cell.

For simplicity, we assume that the transmission cost is proportional to the distance in terms of the number of hops between the source and destination mobility agents such as HA, MAP, and MN. Using the proportional constant δ_U, each cost of location update can be rewritten as follows:

$$C_{HA} = a_h + 2(l_{hm} + l_{mn})\delta_U \tag{1}$$

$$C_{MAP} = a_m + 2l_{mn}\delta_U \tag{2}$$

For simplicity, we assume that each domain has the same number of cells κ. The probability of performing a home registration at movement m is:

$$P_h^m = \frac{N-\kappa}{N-1} \cdot \left(\frac{\kappa-1}{N-1}\right)^{m-2}, where\ 2 \le m < \infty \tag{3}$$

It can be shown that the expectation of M is:

$$E[M] = \sum_{m=2}^{\infty} mP_h^m = 1 + \frac{N-1}{N-\kappa} \tag{4}$$

From (1)-(4), we can get the location update cost per unit time as follows:

$$C_{LU} = \frac{E[M]C_{MAP} + C_{HA}}{E[M]T_c} \tag{5}$$

4.2 Packet Delivery Cost in HMIPv6

The packet delivery cost consists of transmission and processing costs. We define the additional costs and parameters used for the performance evaluation of packet delivery cost as follows:

- T_{hm}: The packet delivery transmission cost between the HA and MAP.
- T_{mn}: The packet delivery transmission cost between the MAP and MN.
- v_h: The packet delivery processing cost at the HA.
- v_m: The packet delivery processing cost at the MAP.
- λ_α : The session arrival rate for incoming packet at the MN.
- δ_D: The proportionality constant for packet delivery.
- δ_h: The packet delivery processing cost constant at the HA.
- δ_m: The packet delivery processing cost constant at the MAP.

The packet delivery cost per unit time can be expressed as follows[7]:

$$C_{PD} = v_h + v_m + T_{hm} + T_{mn} \tag{6}$$

We assume that the packet delivery transmission cost is proportional to the distance in terms of the number of hops between the sending and receiving mobility agents with the proportionality constant δ_D. Therefore, T_{hm} and T_{mn} can be represented as T_{hm}=$l_{hm}\delta_D$ and T_{mn}=$l_{mn}\delta_D$. Also, we define proportionality constants δ_h and δ_m. While δ_h is a packet delivery processing constant for lookup time of the binding cache at the HA, δ_m is a packet delivery processing constant for lookup time of the binding cache at the MAP. Therefore, v_h can be represented as v_h=$\lambda_\alpha\delta_h$ and v_m can be represented as v_m=$\lambda_\alpha\delta_m$. Finally, we can get the packet delivery cost per unit time as follows:

$$C_{PD} = (l_{hm} + l_{mn})\delta_D + \lambda_\alpha(\delta_h + \delta_m) \tag{7}$$

Based on the above analysis, we induce the total signaling cost function from (5) and (7):

$$C_{TOT}(\lambda_\alpha, T_c) = C_{LU} + C_{PD} \tag{8}$$

4.3 Total Signaling Cost in PHMIPv6

We define the additional costs and parameters used for the performance evaluation of the location update cost and packet delivery cost as follows:

- α : The steady state probability that an MN is in active mode
- K: The number of the cells in the paging area
- σ_P: The cost for paging a cell
- T_h: The value of the time threshold for the anchor-cell
- T_c: The cell residence time

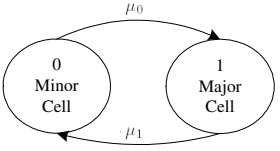


Fig. 2. MN's State Diagram for the Moving Pattern

We define PC_{TOT} and $C_{TOT}^{'}$ to evaluate our proposal. $C_{TOT}^{'}$ denotes the total signaling cost function when an MN is in idle mode. If the MN is in active mode, it operates in exactly the same manner as in HMIPv6. For this reason, we can get a total signaling cost in PHMIPv6, PC_{TOT}, using the sum of C_{TOT} and $C_{TOT}^{'}$. Similarly, $C_{TOT}^{'}$ consists of the location update cost, $C_{LU}^{'}$, and packet delivery cost, $C_{PD}^{'}$ under the MN's idle mode. Also, α denotes the steady state probability that an MN is in active mode. Thus, the steady state probability that an MN is in idle mode can be written as $1 - \alpha$. For simplicity, we assume that the MN's operational mode transition occurs once per macro-mobility. As a result, the total signal cost in PHMIPv6 can be defined as follows:

$$PC_{TOT}(\lambda_\alpha, T_c) = \alpha C_{TOT} + (1 - \alpha) C_{TOT}^{'} \tag{9}$$

$$C_{TOT}^{'} = C_{LU}^{'} + C_{PD}^{'} \tag{10}$$

We assume that an idle MN's movement pattern follows the SMM model[6]. In this case, we define the *major cell* as the cell that an idle MN stays in for a long time. Similarly, we define a *minor cell* as the cell that an idle MN moves among. Also, we assume that the major cell residence time and the minor cell residence time follow the exponential distributions with means $\frac{1}{\mu_1}$ and $\frac{1}{\mu_2}$ ($\frac{1}{\mu_1} \gg \frac{1}{\mu_2}$). We also assume that the total sojourn time among minor cells follows an exponential distribution with the means $\frac{1}{\mu_0}$. Further, we assume that incoming data sessions at an MN follows the poisson distribution with parameters λ_α.

Thus, the idle MN's movement pattern can be analyzed using a birth-and-death process approach as shown in Fig. 2. For simplicity, we assume that μ_0 or μ_1 is not extremely large or small. This implies that the MN does not stay in or move to and from a cell all the time. Thus, T_h is chosen as: $\frac{1}{\mu_1} > T_h > \frac{1}{\mu_2}$. This implies that T_c will be equal to T_h at most times in major-cells, but rarely in minor-cells. Let p_i denotes the steady steady probability at the state i. For a birth-and-death model with $\mu_0 \neq 0$ and $\mu_1 \neq 0$, we can obtain its limiting distribution as follows:

$$\mu_0 p_0 = \mu_1 p_1 \tag{11}$$

Also, we require the conservation relation to hold for state probabilities:

$$\sum_{i=0}^{\infty} p_i = 1 \tag{12}$$

For simplicity, we assume that the idle MN has only one anchor-cell during its stay within a MAP domain. We denote $P_1(T_h)$ and $P_2(T_h)$ as the probabilities that the residence time is larger than T_h in a major cell and in a minor cell, respectively. Since the major cell residence time and the minor cell residence time follow an exponential distribution, we derive $P_i(T_h)$ as follows:

$$P_i(T_h) = e^{-\mu_i T_h} (i = 1, 2) \tag{13}$$

Approximately, the average number of minor cell crossing in state 0 can be presented as $\frac{\mu_2}{\mu_0}$. Thus, we denote the $E[LU]$ as the average number of BU messages while an MN is in idle mode. Thus, we obtain:

$$E[LU] = 2P_1(T_h) + 2P_2(T_h)\frac{\mu_2}{\mu_0} \tag{14}$$

The number of session arrival rate λ_α during the period $E[M]T_c$, is as follows:

$$\frac{1/E[M]T_c}{1/\lambda_\alpha} = \frac{\lambda_\alpha}{E[M]T_c} \tag{15}$$

If the anchor-cell is valid, the number of paged cell is zero. Otherwise, it is K. We denote $E[P]$ as the average of paged cells. Approximately, we get

$$E[P] = p_1[(1 - P_1(T_h)) \cdot K] + p_0[(1 - P_2(T_h)) \cdot K] \tag{16}$$

From Eq. (11) to (16), the total signaling cost for paging can be written as follows:

$$C_p = \sigma_P(\frac{\lambda_\alpha}{E[M]T_c})E[P] \tag{17}$$

Therefore, the total signaling cost for location update per unit time when the MN is idle can be written as follows:

$$C'_{LU} = \frac{E[LU]C_{MAP} + C_{HA}}{E[M]T_c} \tag{18}$$

The PHMIPv6 requires additional control signaling messages compared to the HMIPv6 such as paging request and paging replay. Therefore, the packet delivery cost for the additional control messages can be written as $(1 + C_p) \cdot T_{mn}$ when an idle MN stays in a minor cell. In the case of a major cell, the packet delivery cost is same as the HMIPv6. Thus, the total signaling cost for packet delivery per unit time under MN's idle mode can be written as follows:

$$C'_{PD} = p_0 C_{PD} + p_1(C_{PD} + (1 + C_p)T_{mn}) \tag{19}$$

In addition, we consider the worse case scenario that an idle MN has no anchor-cell. In this case, we get a total signaling cost (i.e., PHMIPv6(no anchor-cell)) for location update and packet delivery per unit time as follows:

$$C'_{LU} = \frac{C_{MAP} + C_{HA}}{E[M]T_c} \tag{20}$$

$$C'_{PD} = C_{PD} + (1 + \sigma_P(\frac{\lambda_\alpha}{E[M]T_c})K) \times T_{mn} \tag{21}$$

Table 1. Performance Analysis Parameter

Parameter	Value	Parameter	Value	Parameter	Value
N	100	κ	$10 \sim 50$	σ_P	$0.2 \sim 10$
a_h	30	a_m	20	λ_α	$0.001 \sim 10$
T_c	0.01 - 100	δ_D	0.2	δ_U	15
T_h	$0.01 \sim 1$	δ_m	0.2	δ_h	1
μ_0	3	μ_2	40	K	$1 \sim 30$

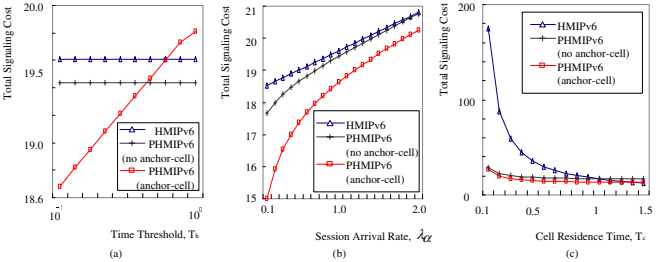


Fig. 3. Effect of T_h, λ_α, and T_c on the Total Signaling Cost

4.4 Analysis of Results

In this section, we demonstrate some numerical results. Table 1 shows some of the parameters used in our performance analysis that are discussed in [4,7]. For simplicity, we assume that the distance between mobility agents are fixed and have the same number. Fig. 3 (a) shows the effect of time threshold T_h on total signaling cost for $\lambda_\alpha = 1$ and $T_c = 1$. As shown in Fig. 3 (a), the total signaling cost for PHMIPv6 increases as the value of time threshold T_h increases. For small values of T_h (i.e. $T_h \leq 0.6$), the performance of the PHMIPv6 is better than that of the HMIPv6. Fig. 3 (b) shows the effect of session arrival rate λ_α on the total signaling cost with cell residence time T_c. As shown in Fig. 3 (b), total signaling cost increases as session arrival rate λ_α increases. We can see that the performance of the PHMIPv6, on the whole, results in the lowest total signaling cost compared with the HMIPv6. For small values of λ_α, performance of the PHMIPv6 is better than that of the HMIPv6. These results are expected because our proposal tries to reduce the number of BU messages when an MN is idle. Therefore, our proposal has a superior performance to the HMIPv6 when λ_α is small. Fig. 3 (c) shows the effect of cell residence time T_c total signaling cost for the fixed value of the session arrival rate λ_α. As residence time T_c increases, total signaling cost decreases. For small values of T_c, the performance of the PHMIPv6 is also better than that of the HMIPv6. This is because when T_c is small, the location update cost is high, which generates high signaling cost in the HMIPv6. However, signaling overhead is not incurred in our proposal when the MN is in idle mode. Based on the above analysis, we come to know that the performance of the PHMIPv6 is better than that of

the HMIPv6, when the session-to-mobility ratio value of the MN is low. So, we conclude that our proposal achieves significant performance improvements by eliminating unnecessary BU messages when an MN is in idle mode.

5 Conclusions

In this paper, we proposed a new IP paging protocol for a hierarchical Mobile IPv6 in IP-based cellular networks. Typically, P-MIP is designed to reduce location update signaling cost. However, it requires additional signaling cost for paging and incurs high paging delays associated with message delivery if the CN sends packets to idle MNs. To reduce signaling cost for paging and delay, we observe that the MN's movement pattern follows SMM model[6]. In our proposal, each MN has the anchor-cell that the MN stays in longer than a certain time threshold T_h. Each MN sends a BU message to update its location to the MAP whenever it enters the anchor-cell or leaves it even if the MN does not communicate with others. Thus, our proposal can reduce the total signaling cost for paging cost and delays when an MN stays in the anchor-cell. Analytical results using the discrete analytic model shows that our proposal can have superior performance to the HMIPv6 when the session-to-mobility ratio value of the MN is low.

References

1. D. B. Johnson and C. E. Perkins. Mobility support in IPv6, *IETF* 2003.
2. Castelluccia, C. Extending mobile IP with adaptive individual paging: a performance analysis. Proc of Fifth IEEE Symposium on Computers and Communications(ISCC 2000); 113–118
3. X. Zhang, J. Castellanos, and A. Campbell, Design and performance of Mobile IP paging. *ACM Mobile Networks and Applications (MONET)* March 2002;7(2).
4. A dynamic anchor-cell assisted paging with an optimal timer for pcs networks Yang Xiao. *IEEE Communications Letters* 2003;7(8);358–360.
5. Hsham Soliman, Claude Castelluccia, Karim El-Malki, Ludovic Bellier. Hierarchical MIPv6 mobility management, IETF 2003.
6. Yu-Chee Tseng, Lien-Wu Chen, Ming-Hour Yang, Jan-Jan Wu, A stop-or-move mobility model for PCS networks and its location-tracking strategies, *Journal of Computer Communication* 2003;26(12);1288–1301.
7. Jiang Xie, Akyildiz, I.F. A novel distributed dynamic location management scheme for minimizing signaling costs in Mobile IP. *IEEE Transactions on Mobile Computing* 2002;1(3);163–175.

Security Enhanced WTLS Handshake Protocol*

Jin Kwak[1], Jongsu Han[1], Soohyun Oh[2], and Dongho Won[1]

[1] School of Information and Communication Engineering,
Sungkyunkwan University,
300 Chunchun-Dong, Jangan-Gu, Suwon, Gyeonggi-Do, 440-746, Korea
{jkwak,jshan,dhwon}@dosan.skku.ac.kr
[2] Division of Computer Science,
Hoseo University, Asan, Chuncheongnam-Do, 336-795, Korea
shoh@dosan.skku.ac.kr

Abstract. WAP is the protocol that is a secure data communication for the wireless environments developed by the WAP Forum. WTLS(Wireless Transport Layer Security) is the proposed protocol for secure communication in the WAP. The purpose of WTLS is to provide secure and efficient services in the wireless Internet environment. However, the existing WTLS handshake protocol has some security problems in several active attacks. Therefore, in this paper, we analyze the securities of the existing protocol, and then propose a security enhanced WTLS Handshake protocol.

Keywords: WAP, WTLS, Handshake protocol, active attack

1 Introduction

Recently, mobile communications have been increasing and many users use the mobile devices in the wireless Internet environments. Accordingly, we are interested with the concerns of WAP(Wireless Application Protocol). WAP is a framework for developing applications to run over wireless networks. WAP protocol consists of 5 layers, such as WDP(Wireless Datagram Protocol), WTLS(Wireless Transport Layer Security), WTP(Wireless Transaction Protocol), WSP(Wireless Session Protocol), and WAE(Wireless Application Layer).

WTLS is the security protocol of the WAP and it operates over the transport layer. The primary goal of WTLS is to provide privacy, data integrity, and to perform the client and the server authentication. WTLS consists of four protocols, such as a Handshake, Alert, ChangeCipherSpec, and Application Data. The Handshake protocol is used to negotiate the security parameters between the client and the server. The Alert protocol is responsible for error handling. If either party detects an error, that party will send an alert containing the error. The ChangeCipherSpec protocol is the protocol that handles the changing of

* This work was supported by the University IT Research Center Project by the MIC (Ministry of Information and Communication) Republic of Korea.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 955–964, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the cipher. This is established while the security parameters are being negotiated during the Handshake.

The WTLS Handshake protocol performs the key distribution phase between the client and the server. However, the existing Handshake protocol is vulnerable to several active attacks, and it can act as fatal weaknesses in the wireless environment. Therefore, in this paper, we analyze the securities of the existing Handshake protocol, and then propose a security enhanced WTLS Handshake protocol, which is secure against active attacks [1,2,3,4].

This paper is organized as follows, in section 2, we explain the existing Handshake protocol and the security analysis of this. Then in section 3, we propose a security enhanced WTLS Handshake protocol, which is secure against active attacks. In section 4, we explain the securities and the characteristics analysis of the proposed protocol. Finally, we will make our conclusions in section 5.

2 Existing WTLS Handshake Protocol

The cryptographic parameters in a secure session are produced by the Handshake. These parameters include attributes such as protocol version, cryptographic algorithms, the method of authentication, and key techniques to generate a shared secret. They agree on secure session information consist of the following items when the client begins the communication with the server [3,5].

- *Session identifier* : arbitrary sequence generated by the server
- *Protocol version* : the version of WTLS protocol
- *Peer Certificate* : certificate of the peer, it can be NULL
- *Compression Method* : algorithm used to compress data
- *Cipher Spec* : specification of the Bulk cipher algorithm and MAC algorithm.
- *Master Secreet* : 20-bytes secret value shared between client and the server
- *Sequence Number Mode* : sequence number used in the connection.
- *Key Refresh* : show an update cycle of a key block
- *Is Resumable* : Flag indicating whether the session can be used to initiate new connection

2.1 Operation of Handshake Protocol

The following definitions and notations are used in this paper, and fig.1 show that the existing Handshake protocol.

· V : is the version of WTLS · E : end entity(client or server)
· SID : session ID · $Cert_E$: certificate of the server or client
· $SecNeg_{C/S}$: information of key exchange suit, cipher suit, key fresh, etc
· K_p : pre-master secret · K_m : master-secret
· $h(\cdot)$: one-way hash function · $u_{m(C/S)}$: middle hash value
· $\pi(\cdot)$: function to output a x coordinate of the point($\pi((x, y)) = x$)
· $R_{C/S}$: random number chosen by client and the server

- $\|$: concatenation
- $\oplus$: exclusive XOR
- x_{CA} : secret key of CA
- P_{CA} : public key of CA
- $x_{C/S}$: long-term secret key of the server and client
- $P_{C/S}$: long-term public key of the server and client
- $x_{T(C/S)}$ and $P_{T(C/S)}$: temporary-secret and public key
- $CI_{C/S}$: information of related certificates include public key, includes *serial number*, $P_{C/S}$, *identifier of signer and issuer*, *etc*
- *kdf* : key derivation function defined ANSI X9.63 [6].
- G : generator of Elliptic Curve
- n : order of generator G
- k_{CA} : random value chosen by CA, where $k_{CA} \in [1, n)$
- $k_{C/S}$: random value chosen by the client and the server, where $k_{C/S} \in [1, n)$
- $T'_{C/S}$: coordinate x of point $T_{C/S}$

1. The client and the server sends the information, such as the random value, WTLS version, SID, and algorithms through a ClientHello and ServerHello message to each other.
2. The server sends the certificate($Cert_S$), CertificateRequest, and ServerHelloDone message to the client, then waits for the response of the client.

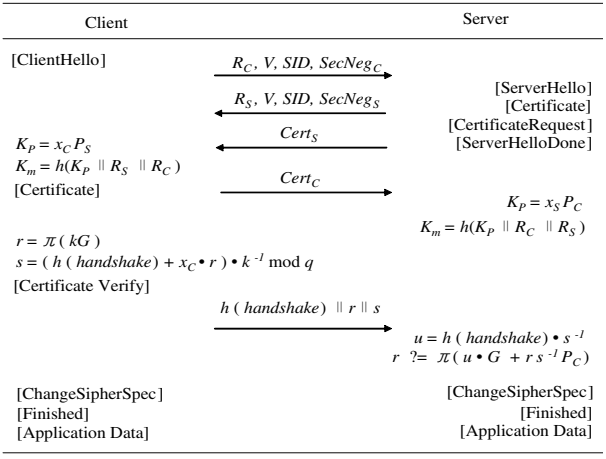


Fig. 1. Existing WTLS Handshake protocol

3. The client verifies received $Cert_S$, and computes pre-master secret K_p with the server's public key. Then he computes master secret K_m with pre-master secret, receiving a random value R_S, R_C, and a one-way hash function. He sends the certificate($=Cert_C$) to the server.
4. The server verifies the received $Cert_C$, and computes pre-master secret K_p with the client's public key. Then he computes master secret K_m with pre-master secret, random value R_S, R_C, and a one-way hash function.

5. The client computes kG using randomly chosen k, and computes r of coordinate x. Then he signs s to the transmitted Handshake message, and then he sends CertificateVerifiy message($= h(Handshake) \parallel r \parallel s$) to the server. The server verifies the client's received signature.
6. Finally, the client and the server send ChangeCipherSpec and Finished messages to each other. The handshake protocol can begin the exchange of application data.

2.2 Security Analysis of Handshake Protocol

Cryptographic attacks are divided into active and passive attacks. In this paper, we will consider active attacks only. An active attack is one where the adversary attempts to delete, add, or in some other way, alter the transmission on the channel. An active attacker threatens data integrity and authentication as well as confidentiality. The discussion in this paper limits considerations to attacks as follows [7,8].

[**Definition 1**] ***Active Impersonation attack (AI)*** : An adversary enters a session with a legitimate entity, impersonates another entity, and finally computes a session key between entities.

Existing Handshake protocol using a long-term key pair for entity authentication and verifies the certificate, therefore, an adversary cannot perform AI. That is to say, if an adversary tries to impersonate, as an entity doesn't know the secret key of the entity, she has to solve the problem of EC-DH(Elliptic Curve(EC) cryptography based Diffie-Hellman(DH) key exchange). Also, an adversary cannot generate a valid signature correspondence to the public key in the transmitted certificate, so the difficulty of the AI attack is the equivalent to the difficulty of the EC-DSA(Elliptic Curve cryptography based Digital Signature Algorithm(DSA)) problem.

[**Definition 2**] ***Forward Secrecy (FS)*** : The secrecy of the previous session keys(master secret K_m in this paper) established by a legitimate entity is compromised. A distinction is sometimes made between the scenario in which a single entity's secret key is compromised(half Forward Secrecy) and the scenario in which the secret key of both entities are compromised(full Forward Secrecy)[8].

The master-secret of existing Handshake protocol is generated by a pre-master secret and a random value $R_{C/S}$. Accordingly, a master-secret can be easily computed by an adversary who knows the client or the server's secret key. Therefore, the existing handshake protocol does not provide any forward secrecy.

[**Definition 3**] ***Key-Compromise Impersonation attack (KCI)*** : Suppose the secret key of an entity is disclosed. Clearly an adversary that knows this value can now impersonate an entity, since it is precisely this value that identifies the entity. In some circumstances, it maybe desirable that this loss

does not enable the adversary to impersonate the other entity and share a key with the entity as the legitimate user.

In the existing Handshake protocol, an adversary can impersonate the client and the server if the secret key of the client and the server is compromised. Therefore, the existing Handshake protocol does not have a Key Compromise Resilience. An adversary sends a random number R_A to the client, and she receives R_C from the client. Then, an adversary computes a pre-master secret($K_p = x_C \cdot P_S$) and a master-secret($K_m = h(K_p \parallel R_C \parallel R_A)$).

[Definition 4] ***Known Key Secrecy (KKS)*** : A protocol should still achieve its goal in the face of an adversary who learns some other session keys. A protocol is said to provide a known key security if this condition is satisfied. There are two kinds of known key attacks, such as $Known\ Key\ Passive\ attack(KKP)$ and $Known\ Key\ Impersonation\ attack(KKI)$(for details, see [7])

WTLS handshake protocol is not secure against KKP and KKI if compromised by the previous pre-master secret. In KKP, an adversary can compute the master-secret with a pre-master secret and a random number, since the pre-master secret consists of a long-term secret key and public information. So, an adversary can easily obtain the master-secret of the current session. Also KKI is the same as the above, and an adversary can easily compute the master-secret if she has the previous pre-master secret and enters the session.

3 Security Enhanced WTLS Handshake Protocol

In this section, we propose security enhanced WTLS Handshake protocol which is secure against several of the defined active attacks(in section 2.2). The proposed protocol needs the procedure of the issuing certificate, so we describes this shortly in this paper.

An end entity(server or client) performs the identity authentication procedure to CA(Certification Authority). He confirms the CI_E issued by CA, which includes long-term public key P_E, serial number, ID of the signer and the issuer, the validity time, etc. CA chooses a random value k_{CA}, and computes $r_E(= k_{CA}G)$ and $r'_E(= \pi(r_E))$. Then, CA generates a signature $s_E(= x_{CA} \cdot h(CI_E \parallel r'_E) + k_{CA} \bmod n)$ using the secret key x_{CA}, and issues a certificate $Cert_E = ((r'_E \parallel s_E))$. Next, CA generates a session key $K(= kdf(x_{CA} \cdot P_E))$ using a key derivation function, and sends the encrypted data $C(= E_K(Cert_E \parallel CI_E))$ to the end entity. After receiving C, the end entity computes the session key $K = (kdf(x_E \cdot P_{CA}))$ and gets the $Cert_E$ and CI_E from the decrypted $C(= D_K(E_K(C)))$. An end entity computes $u_{mE}(= h(CI_E \parallel r'_E))$ and verifies the CA's signature($s_E G - u_{mE} P_{CA} = r_E,\ r'_E \stackrel{?}{=} \pi(r_E)$). Finally, the end entity stores s_E secretly, and uses CI_E and r_E as certificate values.

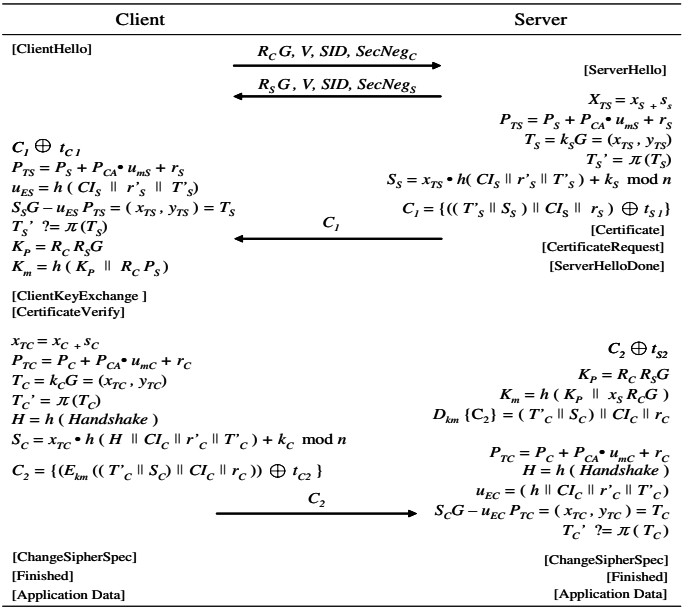


Fig. 2. Security Enhanced WTLS Handshake protocol

3.1 Operations

The operations of the propose security enhanced Handshake protocol are as follows. Fig.2 show that the propose security enhanced WTLS Handshake protocol.

1. The client and the server sends the information to each other, and the server computes a temporary-secret key x_{TS} using long-term secret key x_S and s_s.
2. The server computes a temporary-public key P_{TS} using his long-term public key P_S, CA's public key P_{CA}, r_s, and middle-hash value $u_{mS}(= h(CI_s \parallel r'_s))$. Next, the server chooses random value k_S, then he computes T_S and T'_S.
3. The server computes his signature S_s using x_{TS}, and the server sends C_1, CertificateRequest message, and ServerHelloDone message. Then the server waits for the response of the client. Where t_{S1} is the synchronized time with t_{C1}, and t_{S1} is the computed time of the C_1.
4. The client computes $C_1 \oplus t_{C1}$, and confirms the long-term public key of the server using received CI_S. Then, he computes the temporary-public key P_{TS} using CA's public key P_{CA}, r_S, middle-hash value u_{mS}.
5. The client computes u_{ES} and verifies the server's signature$(S_S G - u_{ES} P_{TS} = T_s,\ T'_S\ ?=\ \pi(T_S))$ using the server's temporary-public key P_{TS}.
6. The client computes a pre-master secret K_p using received $R_S G$ and his R_C, and then computes a master-secret K_m using the pre-master secret, the server's public key, and R_C.

7. The client computes a temporary-secret key x_{TC} using his long-term secret key x_C and s_c, and computes a temporary-public key P_{TC} using his long-term public key P_C, CA's public key P_{CA}, r_C, and middle-hash value $u_{mC}(= h(CI_C \parallel r'_C))$.
8. The client chooses a random value k_C, and computes T_C and T'_C. Then he computes a signature S_C using his temporary-secret key x_{TC}. The client sends the encrypted data C_2 to the server. Where t_{C2} is the synchronized time with t_{S2}, and t_{C2} is the computed time of the C_2.
9. The server computes $C_2 \oplus t_{S2}$, where t_{S2} is the received time of the C_2 from the client, then the server computes the pre-master secret K_p using $R_C G$ and R_S. Next, he computes the master-secret K_m using his secret key x_S and $R_C G$.
10. The server decrypts the C_2, and then he obtains the client's signature and certificate values.
11. The server computes the client's temporary-public key P_{TC} using P_C, CA's public key P_{CA}, r_C, and u_{mC}. Next, the server computes u_{EC}, and verifies the signature$(S_C G - u_{EC} P_{TC} = T_C,\ T'_C \stackrel{?}{=} \pi(T_C))$ of the client using computed P_{TC}.
12. Finally, the client and the server sends the ChangeCipherSpec and the Finished message to each other.

4 Analysis

4.1 Security Analysis

Proposed protocol is based on the difficulty of EC-DH and EC-DSA, therefore an attacker computes the master-secret using public information and transmission information, which is the equivalent to solve the difficulty of ECC based problems [9].

4.1.1 Difficulty of AI

Since the proposed protocol enables the server and the client to authenticate with the signature generated by their own secret key as well as the certificates issued by CA. It does not allow AI attack in which any attacker who does not know the secret key of each entity, and masquerades as an other entity, participates in the protocol and successfully shares a master-secret with the valid entity. The proposed protocol uses the temporary-public and secret key, therefore, an attack is impossible since the attacker should not know the secret key of CA, and randomly selected k_{CA} to computed valid key pair.

4.1.2 Difficulty of FS

The master-secret of the proposed protocol, which the client computes with the public key of the server's P_S, random value R_C selected by himself, and the pre-master secret K_p. If the secret key of the client is exposed and the attacker obtains this, the attacker cannot compute the previous master-secret. The attacker should obtain the R_C, which is randomly selected by the client in

order to compute the master-secret, and since it is as difficult as solving the ECC problem, the attack is impossible. The server computes $R_C G$, and the random point value which is transmitted from the client and the pre-master secret, as well as its own secret key. If the secret key of the server is exposed, the attacker can compute part of the master-secret, but cannot compute the pre-master secret.

Therefore, an attacker cannot compute a valid master-secret. The attacker should get R_C or R_S from the random point value, in order to compute the master-secret, and since it is as difficult as solving the ECC problem, therefore the attack is impossible. Accordingly, the proposed protocol satisfies a full forward secrecy since the master-secret is secure even if the secret key of each entity is exposed.

4.1.3 Difficulty of KCI

In the proposed protocol, the attacker cannot masquerade as the client to the server or masquerade as the server to the client, even if the long-term secret key of the client is exposed. The proposed protocol does not use a long-term key as it is, but generates and uses a temporary-key pair using S_C. It secretly keeps it among the value of certificates issued by CA, in order to prevent this attack. Accordingly, the attacker cannot get a temporary-secret key even if it obtains the long-term key of the client, if he does not know the S_C.

In addition, since the attacker cannot know the r_C transmitted from CA even if the temporary-key is exposed, he cannot compute a valid S_C and C_2. Therefore, the attacker cannot masquerade as a valid client even if he obtains the long-term secret key x_C or the temporary-secret key. Since the server performs the session by computing the temporary-key pair which is the same as the client, the attacker cannot masquerade as anyone, even if the long-term secret key of each entity is exposed. Therefore, the proposed protocol can guarantee security against KCI.

4.1.4 Difficulty of KKS

Since the proposed protocol uses a random pre-master secret for each session in order to compute a master-secret, it cannot give any advantage in obtaining the master-secret of the current session even if the master-secret and the previous transmission information are exposed. Therefore, the difficulty of a KKP attack is the same as the passive attacker who has no information on a previous session. It is also impossible for a KKI attack, because an adversary cannot go directly or participate in the session, or to masquerade as the server or the client using the transformation information of the current session, or the master-secret information, or the transformation information of the previous session, or to set the master-secret.

4.2 Characteristics

In this subsection, we analyze the efficiency of the proposed protocol, and compare it with the existing protocol. To the efficiency analyze of the proposed protocol, we will considered the properties of n-pass, entity authentication, key

Table 1. Characteristics of the proposed protocol and existing protocol

Properties	Existing protocol	Proposed protocol
n-pass	5	4
entity authentication	one-way	mutual
key confirmation	×	one-way
explicit key authentication	×	one-way
user anonymity	×	◯

confirmation, key authentication, key freshness, and user anonymity[10,11,12]. Table.1 shows the characteristics of the proposed protocol and the existing protocol.

The existing protocol is a 5-pass protocol, but the proposed protocol is a 4-pass protocol. Therefore, the proposed protocol is more efficient than the existing protocol. The existing protocol provides a one-way entity authentication, because the client generates a signature and then the server verifies it. In contrast, the proposed protocol provides mutual entity authentication since the server and the client generates a signatures and verify each other.

The existing protocol does not provide key confirmation and explicit key authentication, because there is no key confirmation phase. Besides, the proposed protocol can provide key confirmation. The server decrypts received data encrypted by the client's master-secret, thus can provide key confirmation. Therefore, the client provides explicit key authentication to the server.

Both protocols provide implicit key authentication. The server and the client using the long-term key when they compute the master-secret, are assured that no other entity aside from a specification identified second party could possibly compute the master-secret. In addition, both protocols provide a mutual key freshness, since they compute master-secrets with random values chosen by each other.

The existing protocol is vulnerable to privacy, because the certificate of the client is transmitted in plaintext type. In contrast, the proposed protocol can provide the user's anonymity, since the certificate of the client is transmitted in ciphertext type. Therefore, the proposed protocol cannot expose the identity of the client.

5 Conclusion

In the paper, we analyze the securities and properties of the existing Handshake protocol. As a result, the existing protocol has vulnerabilities to several active attacks. Therefore, we proposed a security enhanced WTLS Handshake protocol secure against active attacks, such as AI, KCI, FS, and KKS. In addition, the proposed protocol can provide mutual entity authentication, one-way key confirmation, mutual key freshness, and user anonymity. The proposed protocol can be implemented efficiently, and provides the additional properties of security and efficiency which have been analyzed in this paper.

References

1. A. Levi and E. Savas, Performance Evaluation of Public-Key Cryptosystem Operations in WTLS Protocol, Proceedings of the 8th IEEE International Symposium on Computers and Communication, ISCC'03, pp. 1245–1250, 2003.
2. T. Dierks and C. Allen, The TLS Protocol version 1.0, IETF RFC 2246, Jan, 1999.
3. WAP Forum, Wireless Application Protocol Wireless Transport Layer Security Specification version 18-FEB-2000, Feb, 2000.
4. G. Radhamani and K. Ramasamy, Security Issues in WAP WTLS Protocol, IEEE 2002 International Conference on Communication, Circuits and Systems and West Sino Expositions, Vol. 1, pp. 483–487, 2002.
5. D. J. Kwak, J. C. Ha, H. J. Lee, H. K. Kim, and S. J. Moon, A WTLS Handshke Protocol with User Anonymity and Forward Secrecy, Mobile Communications: 7th CDMA International Conference, CIC 2002, LNCS 2524, pp.219–230, 2003.
6. ANSI, Public Key Cryptography for the finacial services industry : Key agreement and key transport using elliptic curve cryptography, ANSI X9.63, 2001.
7. S. H. Oh, J. Kwak, S. W. Lee, and D. H. Won, Security Analysis and Applications of Standard key Agreement Protocols, International Conference on Computational Science and Its Applications, ICCSA 2003, LNCS 2668, Part 2, pp. 191–200, 2003.
8. C. Gunther, An Identity-based Key-exchange Protocol, Advances in Crpytology-Eurocrypto'89, LNCS 434, pp. 29–37, 1990.
9. N. Koblitz, Elliptic curve Cryptosystems, Mathematics of Computation, Vol. 48, no. 177, pp. 203–209, 1987.
10. J. P. Yang, W. S. Shin, and K. H. Rhee, An End-to-End Authentication Protocol in Wireless Application Protocol, Information Security and Privacy: 6th Australasian Conference, ACISP 2001, LNCS 2119, pp. 247–259, 2001.
11. W. Zheng, An Authentication and Security Protocol for Mobile Computing, Proceeding of IFIP, pp. 249–257, Sep, 1996.
12. J. S. Go and K. J. Kim, Wireless authentication Protocols Preserving User Anonymity, SCIS 2001, Jan, 2001.

An Adaptive Security Model for Heterogeneous Networks Using MAUT and Simple Heuristics

Jongwoo Chae [1], Ghita Kouadri Mostéfaoui [2], and Mokdong Chung [1]

[1] Dept. of Computer Engineering, Pukyong National University,
599-1 Daeyeon-3Dong, Nam-Gu, Busan, Korea
madcow5980@hanmail.net, mdchung@pknu.ac.kr
[2] Software Engineering Group, University of Fribourg
Rue Faucigny 2, CH-1700 Fribourg, Switzerland
Ghita.KouadriMostefaoui@unifr.ch

Abstract. In this paper, we present an adaptive security model that aims at securing resources in heterogeneous networks. Traditional security model usually work according to a static decision-making approach. However, we may establish a better approach to heterogeneous networks if we use a dynamic approach to construct the security level. Security management relies on a set of contextual information collected from the user and the resource environments, and that infers the security level to enforce. These security levels are dynamically deduced using one of these two algorithms: MAUT and Simple Heuristics.

1 Introduction

Traditional security model usually work according to a static decision-making approach. However, we may establish a better approach to heterogeneous networks if we use a dynamic approach to construct the security level. In the current computing environment, heterogeneous networks are widely available and they have many different properties such as transmission speed, communication media, connectivity, bandwidth, range and etc. Moreover, many types of computing devices are widely used and they have diverse capabilities. To secure this diverse environment, we should adapt several security levels dynamically according to the diverse networks and computing devices. Unfortunately, these characteristics of heterogeneous networks and diverse computing capabilities are dynamically changing by contexts. To cope with this dynamic computing environment, we should make security level more adaptive.

In this paper, we develop an adaptive security model that dynamically adapts the security level according to a set of contextual information such as terminal types, service types, network types, user's preferences, information sensitivity, user's role, location, and time, using MAUT (Multi-Attribute Utility Theory) and Simple Heuristics in order to support secure transactions in the heterogeneous network.

The remainder of the paper is organized as follows: Section 2 discusses related work. Section 3 focuses on the architecture of our system and the theoretical foundations of the used algorithms. Section 4 shows a case study, and Section 5 concludes this paper.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 965–974, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Related Work

From the large panel of definitions of "context," we retain Dey's definition [7]. Context may include physical parameters (type of network, physical location, temperature, etc), human factors (user's preferences, social environment, user's task, etc), and is primarily used to customize a system behavior according the situation of use and/or users' preferences. Context-aware computing is an emerging research field that concerns many areas as information retrieval [22], artificial intelligence [2], computer vision [20], and pervasive computing [8][15][23]. As a consequence many context-aware projects emerged as Cyberguide [11] and SART [1][16]. Even if the list is not exhaustive, most context-aware applications show a strong focus on location [17]. This situation is probably due to the lack of variety of sensors used and the difficulty in sensing high-level contexts, such as in a meeting or intrusion detection.

Considering context in security is a recent research direction. Most of the efforts are directed towards securing context-aware applications. In [4] and [6], Covington et al. explore new access control models and security policies to secure both information and resources in an intelligent home-environment. Their framework makes use of environment roles [5]. In the same direction, Masone designed and implemented RDL (Role-Definition Language), a simple programming language to describe roles in terms of context information [13]. There have also been similar initiatives in [19] and [14]. Interestingly, we observed that all previous work on combining security and context-aware computing follow the same pattern: using contextual information to enrich the access control model in order to secure context-aware applications with a focus on specific applications. The second main observation is that security decisions follow an old-fashioned rule-based formalism which does not consider systems and networks dynamics.

In a recent work [10], Kouadri and Brézillon propose a conceptual model for security management in pervasive environments. The proposed architecture aims at providing context-based authorizations in a federation of heterogeneous resources, and their architecture allows easy plugging of various formalisms for modeling context-based security policies. Our study is inspired from their model with a concrete implementation of the module that manages the context-based security policy.

Kouadri and Brézillon made the first attempt to define a security context [10]. Parts of contextual information are gathered from the pervasive environment where the other part is provided by the requesting user such as its preferences for instance.

Chung and Honavar [3] suggested the negotiating technique across multiple terms of transaction. Determining of security levels may resort to negotiating across multiple terms of transaction, such as terminal types, service types, user's preference, and the level of sensitivity of information.

Section 3 below describes in more detail the types of contextual information our system relies on, the theoretical foundations of MAUT and Simple Heuristics, which deduct more appropriate security level and how this security level will be applied.

3 A Context-Based Adaptive Security Model

This Section describes the physical architecture of our context-based adaptive security model. We consider a set of heterogeneous resources (typically printers, e-learning systems, and computing services) logically grouped into a federation; a university local area network for instance. Figure 1 illustrates the role of the context engine in mediating clients' requests to the protected resources available on the network federation. Users of these resources interact with the services using various means as home computers, handled devices and notebooks with wireless connection. The role of the context engine is to apply the context-based policy by implementing the logic that derives the security level to enforce between a service and its client in each situation (security context). Formally, in our system, a security context is a collection of context elements that are sufficient for applying a given security level. Context elements are predicates involving a number of terms that refer to the user, and computational security context. They include terminal types, service types, user's preference, and the level of sensitivity of information.

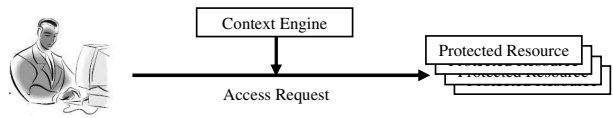


Fig. 1. Overall Architecture

3.1 Contextual Information for the Adaptive Security Model

In order to develop an adaptive protocol which could be used in the diverse environments, we need to define the security level explicitly. The development of such a protocol leads to overcome the limitation of the traditional security model, which sticks to a uniform use of cryptographic techniques, by introducing classification of security level according to domain dependent and independent aspects. Generally, we can classify the security contexts which affect the security level by the two aspects such as domain-independent and domain-dependent contexts.

A mathematical model for the adaptive security system based on contextual information is defined as follows:

$$\begin{aligned} &U = \sum_{i=1}^{n} k_i u_i(x_i),\ (0 \le U \le 1) \\ &SL = \lceil U*10 \rceil / 2,\ (SL = 0, 1, \ldots, 5) \\ &P_{SL} = \{p_0, p_1, \ldots, p_5\} \\ &p_i = \{S_j, A_l, R_m\} \end{aligned} \tag{1}$$

This model determines the adaptive security level to meet the dynamic changes of environmental attributes in the heterogeneous networks. Based on this security level, this model adaptively adjusts the values of the environmental attributes of a security system such as algorithm type, key size, authentication method, and/or protocol type.

p_i consists of a tuple of (S_j, A_l, R_m), determined by the security policy, where S_j is encryption algorithm type, A_l is authentication type, and R_m is protocol type. U is the total utility value, u_i is a utility value of environmental attributes in the heterogeneous networks, and k_i is a scaling constant of the environmental attributes. *SL* represents security level from 0 through 5, where value 0 means we can not utilize the security system. The larger the number is, the stronger the strength of security is. Table 1 shows several algorithm types, and Table 2 illustrates protocol types and authentication methods.

Table 1. Algorithm types (S_j)

S_j	*Symmetric (Key size)*	*Asymmetric (Key size)*	*MAC*
S_0	DES	RSA(512)	MD5
S_1	3DES	RSA(512)	MD5
S_2	3DES	RSA(768)	SHA
S_3	AES(128)	RSA(1024)	SHA
S_4	AES(192)	RSA(1024)	SHA

Table 2. Protocol types (R_m) and Authentication methods (A_l)

R_m	*Protocol types*	A_l	*Authentication methods*
R_0	SPKI	A_0	Password based only
R_1	Wireless PKI	A_1	Certificate based
R_2	PKI	A_2	Biometric based
		A_3	Hybrid methods

3.2 Multi-attribute Utility Theory

Multi-Attribute Utility Theory is a systematic method that identifies and analyzes multiple variables in order to provide a common basis for arriving at a decision. As a decision making tool to predict security levels depending on the security context (network state, the resource's and user's environments, etc), MAUT suggests how a decision maker should think systematically about identifying and structuring objectives, about vexing value tradeoffs, and about balancing various risks. The decision maker assigns utility values to consequences associated with the paths through the decision tree. This measurement not only reflects the decision maker's ordinal rankings for different consequences, but also indicates him relative preferences for lotteries over these consequences [9].

According to MAUT, the overall evaluation $v(x)$ of an object x is defined as a weighted addition of its evaluation with respect to its relevant value dimensions [21][1]. The common denominator of all these dimensions is the *utility* for the evaluator [18]. The utility quantifies the personal degree of satisfaction of an outcome.

The MAUT algorithm allows us to maximize the expected utility in order to become the appropriate criterion for the decision maker's optimal action.

3.3 Simple Heuristics

The Center for Adaptive Behavior and Cognition is an interdisciplinary research group founded in 1995 to study the psychology of bounded rationality and how good decisions can be made in an uncertain world. This group studies Simple Heuristics. The first reason why we use Simple Heuristics is that security level can be decided without user's detailed preference. And the second reason is that it is difficult to predict the preferences of users concerning the attributes of the security level. It is also difficult for even users to determine their preferences quantitatively which is related to the attributes of the security level.

By the way, different environments can have different specific heuristics. But specificity can also be a danger if a different heuristic were required for every slightly different environment, we would need an unworkable multitude of heuristics. Fast and frugal heuristics avoid this trap by their simplicity and enable them to generalize well to new situations. One of fast and frugal heuristics is Take The Best which tries cues in order, searching for a cue that discriminates between the two objects. It serves as the basis for an inference, and all other cues are ignored. Take The Best outperforms multiple regression, especially when the training set is small [12].

3.4 Security Policy Algorithm

We begin by presenting the security policy algorithm that dynamically adapts the security level according to the domain independent properties such as terminal types, and the domain dependent properties such as the sensitivity of information using MAUT and Simple Heuristics. The variables of the algorithms are as follow:

1. domain dependent variables $I = (i_1, i_2, ..., i_n)$: data size, computing power, network type, terminal type, and so on.
2. domain independent variables $X = (x_1, x_2, ..., x_n)$: user attributes, system attributes
3. security level $SL = (0, 1, 2, ..., 5)$: The larger the number is, the stronger the strength is. If SL is 0, we can not utilize the security system.

The overall algorithms for determining adaptive security level are as follows.

[1] [21] describes other possibilities for aggregation.

```
SecurityLevel(securityProblem)
// securityProblem: Determining security level

// Utilization of domain independent properties
   calculate SL by I end
   if SL = 0 then return SL // no use of security system

// Utilization of domain dependent properties
// user select a strategy between MAUT and S. Heuristics
   if MAUT then SL = MAUT(X)
   if Simple Heuristics then SL = TakeTheBest(X);
   return SL; end;
```

```
MAUT(X)
// Determine total utility function by the interaction
// with  the user according to MAUT

   u(x1,x2,…,xn)=k1u1(x1)+k2u2(x2)+… +knun(xn);
// ki is a set of scaling constants
// xi is a domain dependent variable, where ui(x°i)=0,
// ui(x*i)=1,and ki is positive scaling constant for all i

   ask the user's preference and decide ki
   for i = 1 to n
     do ui(xi) = GetUtilFunction(xi);
   end
   return u(x1,x2,…xn); end;
```

```
GetUtilFunction(xi)
// Determine utility function due to users' preferences
// xi is one of domain dependent variables

   uRiskProne : user is risk prone for xi // convex
   uRiskNeutral : user is risk neutral for xi // linear
   uRiskAverse : user is risk averse for xi //concave
   x : arbitrary chosen from xi
   h : arbitrary chosen amount
   <x+h, x-h> : lottery from x+h to x-h

// where the lottery (x*, p, x°) yields a p chance at x*
// and a (1-p) chance at x°
   ask user to prefer <x+h, x-h> or x; // interaction
   if user prefer <x+h, x-h> then
     return uRiskProne;              // e.g. u = b(2^cx-1)
   elseif user prefer x then
     return uRiskAverse;             // e.g. u = blog2(x+1)
   else return uRiskNeutral;  end; // e.g. u = bx
```

```
TakeTheBest(u(x1,x2,…,xn))
// Take the best, ignore the rest
   u(x1,x2,..,xn) : user's basic preferences
// if the most important preference is xi, then only xi
// is considered to calculate SL
// The other properties except xi are ignored

   u(x1,x2,…,xn) is calculated by only considering xi
   SL is calculated by the value of u(x1,x2,…,xn)
   return SL; end;
```

4 A Case Study

In Section 3, we discussed the mathematical foundations for both the MAUT and Simple Heuristics. In this Section, we present a concrete example that makes use of our security management system and that relies on the set of contextual information described in Section 3.

4.1 An Example of Determining a Utility Function in MAUT

For instance, if the utility function $u(x_1, x_2, x_3)$ with three attributes is additive and utility independent, then

$$U(x_1, x_2, x_3) = k_1u_1(x_1)+ k_2u_2(x_2)+ k_3u_3(x_3), \text{ where } u_i(x^o_i) = 0, u_i(x^*_i) = 1, \text{ for all } i \quad (2)$$

where the least preferred consequence, $u_i(x^o_i) = 0$, the most preferred consequence, $u_i(x^o_i) = 1$ for all i.

And then, we ask the decision maker some meaningful qualitative questions about k_i's to get some feeling for their values. For instance, "Would you rather have attribute $\boldsymbol{X_1}$ pushed to x^*_1 than both attributes $\boldsymbol{X_2}$ and $\boldsymbol{X_3}$ pushed to x^*_2 and x^*_3?" A yes answer would imply $k_1 > k_2 + k_3$, which means $k_1 > .5$. We then ask "Would you rather have attribute $\boldsymbol{X_2}$ pushed from x^o_2 to x^*_2 than $\boldsymbol{X_3}$ pushed from x^o_3 to x^*_3?" A yes answer means $k_2 > k_3$. Suppose that we assess $k_1 = .6$, that is, the decision maker is indifferent between (x^*_1, x^o_2, x^o_3) and the lottery $<(x^*_1, x^*_2, x^*_3), .6, (x^o_1, x^o_2, x^o_3)>$, where the lottery (x^*, p, x^o) yields a p chance at x^* and a (1- p) chance at x^o. Then $(k_2 + k_3) = .4$ and we ask "What is the value of p so that you are indifferent between (x^o_1, x^*_2, x^o_3) and $<(x^o_1, x^*_2, x^*_3), p, (x^o_1, x^o_2, x^o_3)>$?" If the decision maker's response is .7, we have $k_2 = p(k_2 + k_3) = .28$

Then, $u(x_1, x_2, x_3) = .6u_1(x_1) + .28u_2(x_2) + .12u_3(x_3)$. Each $u_i(x_i)$ function is determined by the interaction with the user as follows: If a decision maker is *risk prone* then $u_i(x_i)$ is convex function, such as $b(2^{cx} -1)$; else if a decision maker is *risk averse* then $u_i(x_i)$ is concave function, such as $blog_2(x+1)$; else if a decision maker is *risk neutral* then $u_i(x_i)$ is linear function, such as bx; where b, c > 0 constants.

4.2 An Example of Determining Security Policy and Access Policy

Table 3 is a typical example of security policy. In Table 3, x_{att} is the strength of the cipher, x_{auth} is the authentication method, and x_{res} is the level of protection of the resource to which the user is trying to access. The unit of x_{att} is MIPS-Years which is a metric of time to need to break the protected system. *comp* is computing power for message encryption/decryption, *nType* is network type, and *tType* is terminal type, respectively. We need to have a terminal equipped with better than 200 MHz CPU and bandwidth over 100 Kbps to access to the protected resource A. Also we can use PC, PDA, or Cellular phone. User's preference determines the shape of the utility function as discussed in *GetUtilFunction()*, subsection 3.4.

Table 3. An example of security policy

A Security Policy for *Protected Resource A*	
Action	reading
Utility Function	$u(x_{att}, x_{auth}, x_{res}) = k_{att}\, u(x_{att}) + k_{auth}\, u(x_{auth}) + k_{res}\, u(x_{res})$;
Security Contexts	*comp* ≥ 200 MHz; *nType* ≥100 Kbps; *tType* = PC/PDA/Cell;
User's Preference	*uRiskProne* = $2^{2(x-1)}$; *uRiskNeutral* = x; *uRiskAverse* = $\log_2(x+1)$;

Table 4. Conversion table for environmental attributes

utility attribute \ utility value	0.2	0.5	0.8	1.0
x_{att} (MIPS-Years)	$\geq 10^{0.5}$	$\geq 10^{3}$	$\geq 10^{7}$	$\geq 10^{11}$
x_{auth} (Authentication)	Password only	Certificate	Biometric	Hybrid
x_{res} (level of protection)	No	Low	Medium	High

Table 5. An example of access policy

An Access Policy for Protected Resource A
If ((SL ≥ 2) **and** ((Role = administrator) **or** ((Role = user) **and** (Date = Weekdays **and** 8:00 < Time < 18:00)))) **then** resource A can be read
If ((SL ≥ 3) **and** (Role = administrator)) **Then** resource A can be written

Security policy determines the environmental attributes which will be used in the adaptive security level algorithm, constructs the utility function according to the user's preference, and finally determines the security level by using security level algorithm, *SecurityLevel()*. Access policy provides access right or denial to the protected resource according to the security level and user's privilege.

Table 4 is conversion table for environmental attributes whose utility value is mapped from 0 through 1. Each value may be used to calculate the total utility function value.

Table 5 is an example of access policy where reading or writing access right is given to the user according to the security level, user's role, and/or time attributes. *SL* is the lower bound of security level. Any user cannot adopt *SL* lower than 3 for write operation. If the user is administrator and *SL* is higher than 3, then he or she can write.

4.3 The Strengths of the Proposed Model

The strengths of the proposed model are as follows: Firstly, traditional security model usually work according to a static decision-making approach since the same authentication and authorization protocol might be used to the diverse protected resources, for instance. This might result in the waste of system resources, such as

excessive usage of CPU and excessively high network bandwidth. In the proposed model, we can reduce the waste of resources by adaptively applying appropriate cryptographic techniques and protocols according to the characteristics of the resources. Therefore, the proposed model increases efficiency and availability of the resources.

Secondly, in terms of system protection, our model is more secure than traditional one. When the system identifies possibilities of attacks or vulnerabilities of the resources, our model protects the system by adaptively decreasing security level of the resource. When the security level is decreased, the access request might be denied by applying the rule sets of the access policy.

Finally, the traditional security systems can not consider user's security preference. In the contrast, our model can reflect the user's preference. Therefore, the result of the same access request could be quite different although any other contexts are the same.

5 Conclusion and Future Work

In this paper, we presented an adaptive security model that provides adaptive security policies for heterogeneous networks. Adaptability is expressed using a set of contextual information about all the parties included in the interaction, namely, the protected resource, the requesting user, and the network which represents the working platform for the interaction. For each security context, a security level is enforced by the mean of two algorithms; MAUT and Simple Heuristics. Our system has been applied to a university local area network with a set of heterogeneous services as printer services, e-learning systems, etc. Moreover, the proposed architecture could be applied to any network that offers different types of services and resources, in order to provide context-based fine-grained access to these resources.

In the future, we will analyze quantitatively the effectiveness of the proposed adaptive security model through a simulation or a real implementation in the heterogeneous networks.

References

[1] P. Brézillon, et al., "SART: An intelligent assistant for subway control," Pesquisa Operacional, Brazilian Operations Research Society, vol. 20, no. 2, 2002, pp. 247–268.

[2] P. Brézillon, "Context in Artificial Intelligence: I. A survey of the literature," Computer & Artificial Intelligence, vol. 18, no. 4, 1999, pp. 321–340, http://www-poleia.lip6.fr/~brezil /Pages2/Publications/CAI1-99.pdf.

[3] M. Chung and V. Honavar, "A Negotiation Model in Agent-Mediated Electronic Commerce," Proc. IEEE Int'l Symposium on Multimedia Software Engineering, Taipei, Dec. 2000, pp. 403–410.

[4] M.J. Covington, et al., A Security Architecture for Context-Aware Applications, tech. report, GIT-CC-01-12, College of Computing, Georgia Institute of Technology, May 2001.

[5] M.J. Covington, at el., "Securing Context-Aware Applications Using Environment Roles," Proc. 6th ACM Symposium on Access Control Models and Technologies, Chantilly, VI, USA, May, 2001, pp. 10–20.
[6] M.J. Covington, at el., "A Context-Aware Security Architecture for Emerging Applications," Proc. Annual Computer Security Applications Conf. (ACSAC), Las Vegas, Nevada, USA, Dec. 2002.
[7] A.K. Dey, Ph. D. dissertation, Providing Architectural Support for Building Context-Aware Applications, Georgia Institute of Technology, 2000.
[8] K. Henricksen, at el., "Modeling Context Information in Pervasive Computing Systems," Proc. 1st Int'l Conf., Pervasive 2002, Zurich, Springer Verlag, Lecture Notes in Computer Science, vol. 2414, 2002, pp. 167–180.
[9] R.L. Keeney and H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley & Sons, New York, NY, 1976.
[10] G. K. Mostéfaoui and P. Brézillon, "A generic framework for context-based distributed authorizations," Proc. 4th Int'l and Interdisciplinary Conf. on Modeling and Using Context (Context'03). LNAI 2680, Springer Verlag, pp. 204–217.
[11] S. Long, et al., "Rapid prototyping of mobile context-aware applications: The cyberguide case study," Proc. the 1996 Conf. Human Factors in Computing Systems (CHI'96), 1996, pp. 293–294.
[12] L. Martignon and U. Hoffrage, Why Does One-Reason Decision Making Work? In Simple Heuristics That Make Us Smart, Oxford University Press, New York, 1999, pp. 119–140.
[13] C. Masone, Role Definition Language (RDL): A Language to Describe Context-Aware Roles, tech. report, TR2001-426, Dept. of Computer Science, Dartmouth College, May 2002.
[14] P. Osbakk, and N. Ryan, "Context Privacy, CC/PP, and P3P," Proc. UBICOMP2002 – Workshop on Security in Ubiquitous Computing, 2002, pp. 9–10.
[15] A. Rakotonirainy, Context-oriented programming for pervasive systems, tech. report, University of Queensland, Sep. 2002.
[16] The SART Project, http://www-poleia.lip6.fr/~brezil/SART/index.html.
[17] A. Schmidt, et al., "There is more to context than location," Computers and Graphics, vol. 23, no. 6, Dec. 1999, pp. 893–902.
[18] R. Schäfer, "Rules for Using Multi-Attribute Utility Theory for Estimating a User's Interests," Proc. 9th GI-Workshop. ABIS-Adaptivität und Benutzermodellierung in interaktiven softwaresystemen, Dortmund, Germany, 2001.
[19] N. Shankar, D. Balfanz, "Enabling Secure Ad-hoc Communication Using Context-Aware Security Services (Extended Abstract)," Proc. UBICOMP2002 – Workshop on Security in Ubiquitous Computing.
[20] T. M. Strat, et al., Context-Based Vision. Chapter in RADIUS: Image Understanding for Intelligence Imagery, O. Firschein and T.M. Strat, Eds., Morgan Kaufmann, 1997.
[21] D. Winterfeld, von and W. Edwards, Decision Analysis and Behavioral Research, Cambridge, England: Cambridge University Press, 1986.
[22] M. Weiser, "The computer for the 21st Century," Scientific American, vol. 265, no. 3, 1991, pp. 66–75.
[23] S. S. Yau, et al., "Reconfigurable Context-Sensitive Middleware for Pervasive Computing," IEEE Pervasive Computing, joint special issue with IEEE Personal Communications, vol. 1, no. 3, July-September 2002, pp. 33–40.

A New Mechanism for SIP over Mobile IPv6

Pyung Soo Kim[1], Myung Eui Lee[2], Soohong Park[1], and Young Kuen Kim[1]

[1] Mobile Platform Lab, Digital Media R&D Center,
Samsung Electronics Co., Ltd, Suwon City, 442-742, Korea
Phone : +82-31-200-4635, Fax : +82-31-200-3147
kimps@samsung.com
[2] School of Information Technology,
Korea Univ. of Tech. & Edu., Chonan, 330-708, Korea

Abstract. This paper proposes a new mechanism for Session Initiation Protocol (SIP) over Mobile IPv6. In this mechanism, a home agent (HA) on home subnet acts as a redirect server and a registrar for SIP as well as a home router for Mobile IPv6. Thus, a binding cache in the HA contains location information for SIP as well as home registration entries for Mobile IPv6. An access router on foreign subnet acts as only a router that offers a domain name. To implement the proposed mechanism, some messages used in network layer are newly defined, such as a router advertisement, a router solicitation and a binding update. In the proposed mechanism, a mobile node doesn't require dynamic host configuration protocol (DHCP) and thus both home and foreign subnets don't need DHCP servers, unlike existing mechanisms on Mobile IPv4. Analytic performance evaluation and comparison are made, which shows the proposed mechanism is more efficient in terms of delay than existing mechanisms.

1 Introduction

Over the past few years, an important trend is the emergence of voice over IP (VoIP) services and its rapid growth. For VoIP services, the Session Initiation Protocol (SIP) has been standardized by IETF [1] and researched by paper works [2], [3]. SIP is an application layer protocol used for establishing and tearing down multimedia sessions. Meanwhile, mobility support is also becoming important because of the recent blossoming of mobile appliances, such as mobile phone, handheld PC, laptop computer, and the high desire to have seamless network connectivity. To support mobility for IPv4, Mobile IPv4 [4] was designed by IETF. In addition, in recent, to solve the address exhaustion problem and the routing optimization problem with Mobile IPv4, Mobile IPv6 has been standardized by IETF [5] and researched by paper works [6], [7] for IPv6 [8].

Even though the original SIP and its applications did not consider the mobility of the end nodes, there have been ongoing research efforts to support mobility in the current SIP [9]-[11]. These works have been researched on Mobile IPv4 because Mobile IPv6 was not well established until recent. To authors' knowledge, there seems to be no well established result for SIP over Mobile IPv6.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 975–984, 2004.
© Springer-Verlag Berlin Heidelberg 2004

However, as mentioned before, there are the address exhaustion problem and the routing optimization problem in Mobile IPv4. Therefore, mechanisms for SIP over Mobile IPv6 might be required for wireless and mobile communication environments.

In this paper, a new mechanism for SIP over Mobile IPv6 is proposed. In this mechanism, a home agent (HA) on home subnet acts as a redirect server and a registrar for SIP as well as a home router for Mobile IPv6. That is, the HA provides its inherent functions of Mobile IPv6, such as a router advertisement and a home registration for a mobile node (MN). In addition, for SIP, the HA accepts a location registration request of the MN, places the information it receives in this request into a location database, and returns the location information of the MN to a correspondent node (CN). Thus, a binding cache in the HA contains location information for SIP as well as home registration entries for Mobile IPv6. On the other side, an access router on foreign subnet, which will be called a foreign router (FR) hereafter, provides only a router advertisement for the MN.

To implement the proposed mechanism, in this paper, some new messages used in network layer are defined by adding some fields to existing messages in [5]. Firstly, in order that HA and FR offer a subnet prefix and a domain name for the MN, a router advertisement (RA) message is newly defined. Using this RA message, the MN can make a new Uniform Resource Identifier (URI) as well as a home address (HoA) or a care-of address (CoA). Secondly, in order that the MN solicits the HA or FR for the RA with a domain name as well as a subnet prefix, a router solicitation (RS) message is newly defined. Lastly, when the MN changes its subnet and thus makes the CoA and the new URI, to do simultaneously both location registration for SIP and home registration for Mobile IPv6 with the HA, a binding update (BU) message is newly defined.

In the proposed mechanism, the MN doesn't require dynamic host configuration protocol (DHCP) and thus both home and foreign subnets don't need DHCP servers. On the other hand, existing mechanisms on Mobile IPv4 required DHCP for the MN, and used DHCP servers for the MN to get the HoA or CoA and the new URI, as shown in [9]-[11]. In addition, the proposed mechanism provides the efficient optimized routing where speech packets sent by the CN are routed directly to the MN, whereas existing mechanisms could not due to the triangle routing.

Finally, to evaluate the proposed mechanism, the delay to the SIP call setup from the MN's subnet change is computed analytically for a simplified network model. In addition, the comparison between mechanism and existing mechanisms on Mobile IPv4 is made in terms of delay. These analytic performance evaluation and comparison show the proposed mechanism is more efficient in terms of delay than existing mechanisms.

The paper is organized as follows. In Section 2, the network architecture for the proposed mechanism is introduced. In Section 3, some messages in used network layer are newly defined. In Section 4, the basic operation of the proposed mechanism is explained. In Section 5, analytic performance evaluation and comparison are made. Finally, the conclusions are made in Section 6.

2 Network Architecture for Proposed Mechanism

In this paper, a new mechanism for SIP over Mobile IPv6 is proposed for a wireless mobile network as shown in Fig. 1. As shown in Fig. 1, the network considered for the proposed mechanism consists of a mobile node (MN), a correspondent node (CN), a home agent (HA), and a foreign router (FR).

The MN acts as a user agent (UA) for SIP as well as a mobile host for Mobile IPv6. That is, in addition to its inherent functions for Mobile IPv6, the MN creates a new SIP request and generates a response to a SIP request. The CN also acts as a user agent (UA) for SIP as well as a peer node with which a mobile node is communicating for Mobile IPv6.

The HA on home subnet acts as a redirect server and a registrar for SIP as well as a home router for Mobile IPv6. That is, the HA provides its inherent functions of Mobile IPv6, such as a router advertisement and a home registration for a mobile node (MN). In addition, for SIP, the HA accepts a location registration request of the MN, places the information it receives in this request into a location database, and returns the location information of the MN to the CN. Thus, the binding cache in the HA contains location information for SIP as well as home registration entries for Mobile IPv6. On the other side, the FR provides only a router advertisement for the MN.

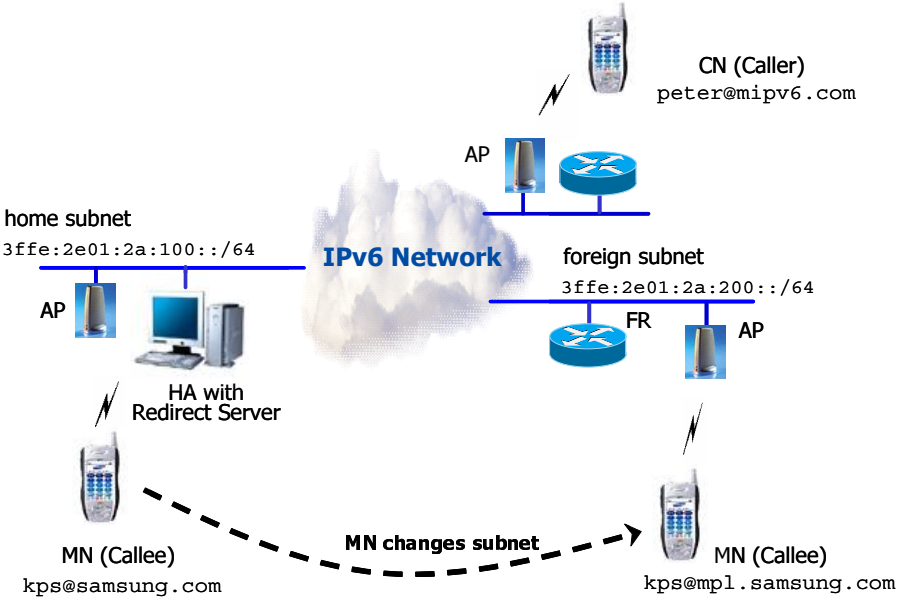


Fig. 1. Network architecture for proposed mechanism

3 New Messages for Proposed Mechanism

In this section, to implement the proposed mechanism, some messages used in network layer are newly defined by adding some fields to existing messages in [5], such as a router advertisement, a router solicitation and a binding update.

3.1 New Router Advertisement Message

In order that HA and FR offer a subnet prefix and a domain name for the MN, a router advertisement (RA) message is newly defined by adding some fields to existing message in [5]. Using this RA message, the MN can make a new Uniform Resource Identifier (URI) as well as a home address (HoA) or a care-of address (CoA).

- Domain Name Flag (D) : This bit is set in a RA to indicate that the router sending this RA also include a domain name of current subnet.
- Reserved : Reduced from a 5-bit field to a 4-bit field to account for the addition of the above bit.
- Domain Name : The domain name of current subnet where the MN is attached. The data in the domain name should be encoded according to DNS encoding rules. For example, `samsung.com` or `mpl.samsung.com`
- Other fields : See [5]

The source address field in the IP header carrying this message is the link-local address assigned to the interface from which this message is sent. The destination address field in the IP header carrying this message is typically the source address of an invoking router solicitation or the all-nodes multicast address.

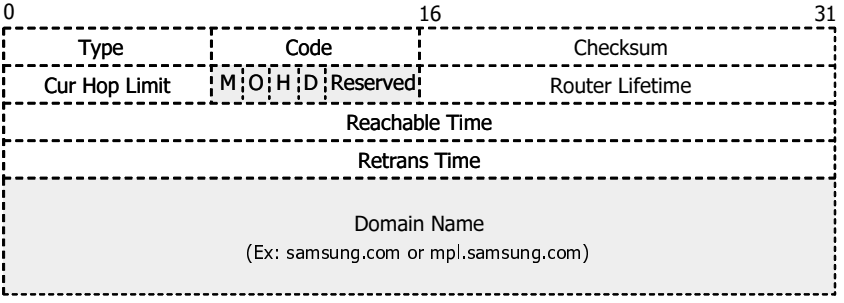


Fig. 2. New router advertisement message

3.2 New Router Solicitation Message

In order that the MN solicits the HA or FR for a router advertisement with a domain name as well as a subnet prefix, a router solicitation (RS) message is newly defined by adding some fields to existing message in [5].

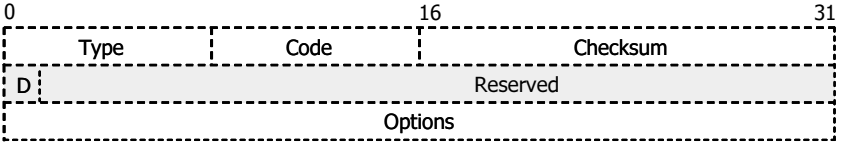


Fig. 3. New router solicitation message

- Domain Name Request Flag (D) : This bit is set in a RS to indicate that the MN requests a domain name of current subnet.
- Reserved : Reduced from a 32-bit field to a 31-bit field to account for the addition of the above bit.
- Other fields : See [5]

The source address field in the IP header carrying this message is the IP address of MN. The destination address field in the IP header carrying this message is typically the all-routers multicast address.

3.3 New Binding Update Message

When the MN changes its subnet and thus makes the CoA and the new URI, to do simultaneously both location registration for SIP and home registration for Mobile IPv6 with the HA, a new binding update (BU) message is newly defined by adding some fields to existing message in [5].

- Option Type : 10 (or any available value)
- New URI : The new URI of the MN. For example, `kps@samsung.com` or `kps@mpl.samsung.com`
- Other fields : See [5]

The source address field in the IP header carrying this message is the CoA of the MN. The destination address field in the IP header carrying this message is the IP address of the HA. This BU message contains the Home Address destination option that has the HoA of the MN.

4 Basic Operation Procedure of Proposed Mechanism

In this section, the basic operation of the proposed mechanism is explained in detail. As shown in Fig. 1, the MN is assumed to be the callee and the CN is assumed to be the caller. It is also assumed that the MN's HoA is `3ffe:2e01:2a:100::10` and URI is `kps@samsung.com` in a home subnet. The CN's URI is assumed to be `peter@mipv6.com`.

When the MN changes its subnet and thus attaches to a foreign subnet, it will receive solicited RA or unsolicited RA from the FR. To receive the solicited RA, the MN sends the RS with setting Domain Name Request Flag (D) to the FR as shown in Fig. 3. Note that this RS would be optional. The RA in Fig. 2 contains

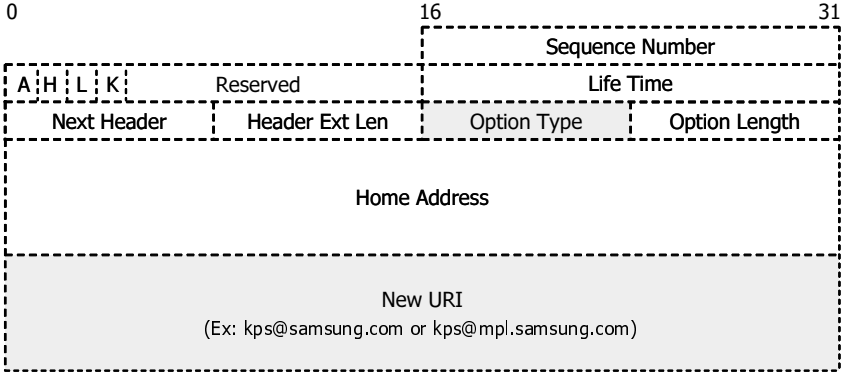


Fig. 4. New binding update message

the subnet prefix for the MN's CoA configuration and the domain name for the MN's new URI configuration. As shown in Fig. 1, in foreign subnet, the address prefix is `3ffe:2e01:2a:200::/64` and the domain name is `mpl.samsung.com`. Then, the MN makes the CoA as `3ffe:2e01:2a:200::10` and the new URI as `kps@mpl.samsung.com`.

To do simultaneously both location registration for SIP and home registration for Mobile IPv6 with the HA, the MN sends the BU with both CoA and new URI to the HA, using the newly defined message in Fig. 4. If the HA accepts the BU, it update its binding cache entry for the MN. If the URI is not changed, the only CoA is updated in the binding cache. Fig. 5 shows the binding cache before and after BU interaction between the HA and the MN. This binding cache is an effective database containing mappings among the original URI, the current URI, the HoA and the CoA.

The CN with `peter@mipv6.com` wants to invite the MN with `kps@samsung.com`. The CN translates the domain name `samsung.com` to a numeric IP address, by a DNS lookup, where the HA may be found. An INVITE request is generated and sent to this HA. Note that the HA does not issue any SIP requests of its own. After receiving a request other than CANCEL, the HA either refuses the request or gathers the MN's current location information from the binding cache and returns a final response of class 3xx. For well-formed CANCEL requests, it returns a 2xx response. Then, when the HA accepts the invitation, it gathers the MN's current location information, such as the HoA `3ffe:2e01:2a:100::10`, the CoA `3ffe:2e01:2a:200::10` and the new URI `kps@mpl.samsung.com`, from the binding cache. Thus, the HA returns a 302 response (Moved Temporarily) with MN's current location information. The CN acknowledges the response with an ACK request to the HA. Then, the CN issues a new INVITE request based on the MN's current URI `kps@mpl.samsung.com`. This request is sent to the MN's CoA `3ffe:2e01:2a:200::10`. In this case, the call succeeds and a response indicating this is sent to the CN. The signaling is completed with an ACK from the CN to the MN. After this call setup, the

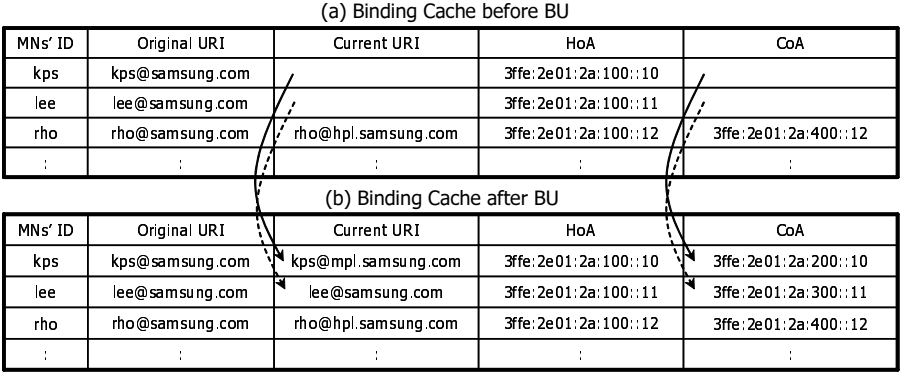


Fig. 5. Binding Cache in HA before/after BU between HA and MN

real speech communication is going on. Fig. 6 shows the basic operation for the proposed mechanism.

In the proposed mechanism, the MN doesn't require dynamic host configuration protocol (DHCP) and thus both home and foreign subnets don't need DHCP servers. On the other hand, existing mechanisms on Mobile IPv4 required DHCP for the MN, and used DHCP servers for the MN to get the HoA or CoA and the new URI, as shown in [9]-[11]. In addition, the proposed mechanism uses the optimized routing between the MN and the CN. That is, in Fig. 1, speech packets sent by the caller are routed directly to the callee. On the other hand, in existing mechanisms [9]-[11], speech packets that are sent by the caller to the callee connected to a foreign subnet are routed first to the callee's HA and then tunneled to the callee's CoA. Therefore, the proposed mechanism might be more efficient in terms of speech delay and resource consumption than existing mechanisms, because, in general, the speech packets will have to traverse fewer subnets on their way to their destination.

5 Analytic Performance Evaluation and Comparison

In this section, to evaluate the proposed mechanism, the delay to the SIP call setup from the MN's subnet change is computed analytically for a simplified network model. In addition, the comparison between mechanism and existing mechanisms on Mobile IPv4 is made in terms of delay. It is assumed that link layer establishment is negligible. A simple network model for analysis is shown in Fig. 7.

The delay between the MN and the FR is assumed to be T_{MF}. The delay between the HA and the FR is assumed to be T_{HF}, and the delay between the HA and the CN is assumed to be T_{HC}, The delay between the CN and the FR is assumed to be T_{CF}. Total delay time is computed to the SIP call setup with the CN from the MN's subnet change. Signals during this total delay can be as follows:

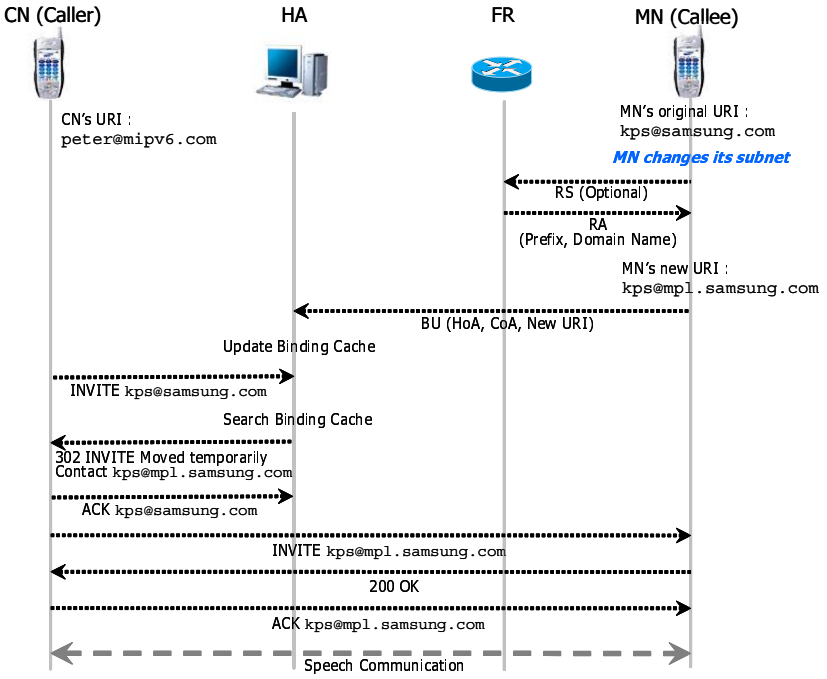


Fig. 6. Basic operation of proposed mechanism

(a) RS and RA interaction
(b) Home registration
(c) DHCP interaction (only for existing mechanisms)
(d) Location registration (only for existing mechanisms)
(e) Redirection interaction
(f) Call establishment

Note in (a) that the MN gets both subnet prefix and domain name in the proposed mechanism, whereas the MN gets only subnet prefix in the existing one. Note in (b) that the home registration is performed with the location registration in the proposed mechanism, whereas these two registrations are performed respectively in existing mechanisms. Thus, (c) and (d) are signals only for existing mechanisms. Note in (e) that the CN gets MN's location information from the HA in the proposed mechanism, whereas the CN gets it from the redirect server in existing mechanisms. Note in (f) that the proposed mechanism uses the optimized routing whereas existing mechanisms use the triangle routing.

As shown in Table 1, the total delay of the proposed mechanism is smaller than that of existing mechanisms. Especially, when the MN is far from its home subnet and near the CN, the result might be remarkable. That is, as T_{HC} and T_{HF} are larger, the total delay of the proposed mechanism is remarkably smaller than that of existing mechanisms. Thus, it can be said that the proposed mechanism is more efficient in terms of delay than existing mechanisms in [9]-[11].

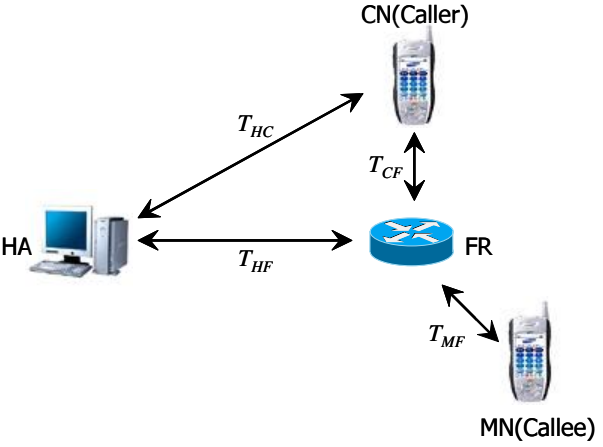


Fig. 7. Overall analytic model

Table 1. Round-trip time for each signaling

Signals	Proposed Mechanism	Existing Mechanisms
(a)	$2T_{MF}$	$2T_{MF}$
(b)	$2(T_{MF} + T_{HF})$	$2(T_{MF} + T_{HF})$
(c)	·	$2T_{MF}$
(d)	·	$2(T_{MF} + T_{HF})$
(e)	$2T_{HC}$	$2T_{HC}$
(f)	$2(T_{CF} + T_{MF})$	$(T_{CF} + T_{MF}) + (T_{MF} + T_{HF} + T_{HC})$

6 Conclusions

In this paper, a new mechanism has been proposed for SIP over Mobile IPv6. In this mechanism, the HA on home subnet acts as a redirect server and a registrar for SIP as well as a home router for Mobile IPv6. Thus, a binding cache in the HA contains location information for SIP as well as home registration entries for Mobile IPv6. The FR acts as only a router that offers a domain name. To implement the proposed mechanism, some messages used in network layer are newly defined such as a router advertisement, a router solicitation and a binding update. In the proposed mechanism, the MN doesn't require DHCP and thus both home and foreign subnets don't need DHCP servers, unlike existing mechanisms on Mobile IPv4. Finally, analytic performance evaluation and comparison have showed that the proposed mechanism is more efficient in terms of delay than existing mechanisms.

References

1. Rosenberg, J. et. al: SIP: Session Initiation Protocol. RFC 3261 (June 2002)
2. Schulzrinne, H., Rosenberg, J.: The session initiation protocol: Internet-centric signalling. IEEE Communications Magazine, Vol.38. (2000) 134–141
3. Robles, T., Ortiz, R., Salvachja, J.: Porting the session initiation protocol to ipv6. IEEE Internet Computing Vol.7. (2002) 43–50
4. Perkins, C.: IP Mobility Support. RFC 2002 (October 1996)
5. Johnson, D. B., Perkins, C. E., Arkko, J: Mobility Support in IPv6. IETF Draft:draft-ietf-mobileip-ipv6-24.txt (July 2003)
6. Costa, X. P., Hartenstein, H.: A simulation study on the performance of Mobile IPv6 in a WLAN-based cellular network. Computer Networks, Vol.40. (2002) 191–204
7. Chao, H. C., Chu, Y. M., Lin, M. T.: The implication of the next-generation wireless network design: cellular mobile IPv6. IEEE Transactions on Consumer Electronics, Vol.46. (2002) 656-663
8. Narten, T., Nordmark, E., Simpson, W.: Neighbor Discovery for IP Version 6 (IPv6). IETF RFC 2461 (December 1998)
9. Moh, M., Berquin, G., Chen, Y.: Mobile ip telephony: mobility support of sip. In: Proc. Int. Conf. on Computer Communications and Networks. (1999) 554–559
10. Seol, S., Kim, M., Yu, C., Lee, J.H.: Experiments and analysis of voice over mobile ip. In: Proc. IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications. (2002) 997–981
11. Kwon, T.T., Gerla, M., Das, S., Das, S.: Mobility management for voip service: Mobile IP vs. SIP. IEEE Wireless Communications Vol.9. (2002) 66–75

A Study for Performance Improvement of Smooth Handoff Using Mobility Management for Mobile IP

Kyu-Tae Oh and Jung-Sun Kim*

School of Electronics, Telecommunication and Computer Engineering, Hankuk Aviation University, 200-1, Hwajeon-dong, Deokyang-gu, Koyang-city, Kyonggi-do, 412-791, Korea
okt@sdsolution.co.kr, jskim@mail.hangkong.ac.kr

Abstract. As a way to improve process rate in case of smooth handoff at the mobile IP, this study introduced function of GFA (Gateway Foreign agent) that performs regional management among FA, and conducted methods to reduce transfer time, which is induced by the register process of FA to HA. Through result of the study, we concluded that using GFA for regional management is more effective than to set buffer in FA in sense of transmission delay. If this result would be utilized in field of modern active internet business, it will effectively respond to the future demand of wireless phone's handoff.

1 Introduction

Recently, internet business is rapidly developing from wire to wireless network. As a way to improve performance of smooth handoff, which is the essential element in internet, we carefully examined how to use GFA for regional management in this paper. Previously introduced smooth handoff methods can be classified as installing buffer in an old FA, in a new FA, and in every FA. GFA used method in the thesis is the applied one that installed buffer in every FA and is the most effective type.

For beginning of the experiment, we proved better efficiency improvement for the type that installed buffer in every FA with other types and later on, we performed an experiment to prove another efficiency improvement when it has the regional management GFA on this type.

In mobile IP, there must be handoff between MAC and IP layer in case of handoff. In that sense, the handoff at MAC layer is to secure reliance of wireless link and the handoff at IP level to provide transparency of MN. [1]

There are two different types of address category in mobile IP; home address and COA (Card of Address). The home address is assigned address from MN to HA from the beginning and tries to connect every host that wants to communicate with MN to the home address. COA cannot use home address that it used to use when MN moves to FA, thus it needs to get an address from FA that is only usable in FA. Such newly assigned address is called COA.

Type of COA is categorized into foreign agent COA and co-located COA. The foreign agent COA uses address of FA as COA and the co-located COA is temporary assigned address from FA. Address assigning uses protocol like DHCP. [2]

* The corresponding author will reply to any question and problem from this paper.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 985–992, 2004.
© Springer-Verlag Berlin Heidelberg 2004

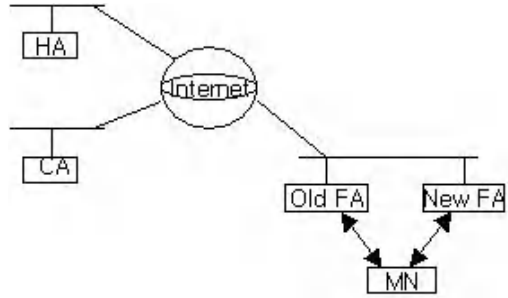


Fig. 1. Network Architecture of MobileIpv6

It supports mobility of MN by using such two addresses and the operation that connects home address and COA of MN is called as Binding. In order to seize the location of MN in FA, it sends an advertising message to MN and does neighbor discovery and through this process, MN gains COA by the advertising. [3].

When MN sends binding information to HA, HA updates the binding information and connects home address and COA. In this case, if CN wants the connection with MN as HA then HA responses the address of COA to CN, and CN connects to MN as COA in FA.

If MN does handoff as another FA (New FA) while CN and MN are communicating, it receives agent advertisement message from the new FA and gains a new COA. Once the COA is registered in HA, HA sends binding message to every host that has binding entry of MN and informs the location change of MN. [4] Through this work, CN can communicate with MN that is in a new FA.

When handoff occurs in mobile IP, it is impossible for CN to communicate with MN in a new FA till a new COA is registered in HA and until the registration of COA to HA be done, it tries to connect to an old FA while there is handoff. In this process, every datagram that has sent to the old FA is deleted and it causes transmission delay.

Thus in this paper as a way to minimize the transmission delay in case of handoff, we analyzed and developed methods to utilize GFA that manages FA, and based on the study we carefully figured out the most effective way for smooth handoff in mobile IP.

2 Types of Handoff Protocol That Install Buffer in Every FA

The fast handoff technique is often used for mobile IPv6 and in order for this fast handoff, it has to have and then manage binding information about MN in the binding cache entry of CN. When there is handoff because of MN's change in position, COA in a new FA would be connected to HA. At this time, CN regard MN is presently in the previous FA, thus send datagram to the previous FA, and the sent data gram is deleted in the previous FA. Once handoff is set, HA completes updating binding and CN transmits the datagram that was deleted in the previous FA to a new FA in MN. This process causes the waste of traffic and eventually a transmission delay happens.

To avoid such delay in transmission, there are two ways to install buffer; one is to an old FA and the other is to a new FA. Both ways can reduce transmission delay, but

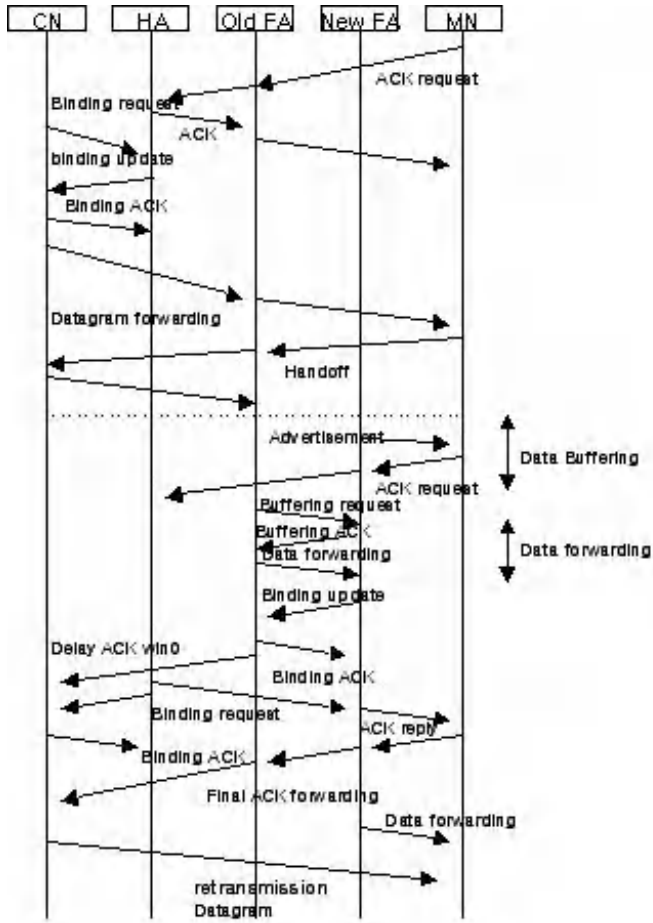


Fig. 2. Handoff protocol that install buffer in every FA

they also have several defects. The suggested way in this paragraph is type to compensate the defects that the listed above have. When MN directly returns to an old FA from a new FA, the buffered datagram in the new FA is ignored and the buffered datagram in the old FA can be transmitted to MN.

Also when it receives datagram that goes to MN from an old FA during handoff, it saves received datagram in its buffer as well as transmits datagram to a buffer of a new FA. This way has to have installed buffer in both an old and a new FA, so it is little costly. However, the buffer's capacity is relatively small and numbers of MN that one FA monitors are generally less than 7, thus amount of datagram does not require a big size buffer so it is economic in this sense. This way is usually effective when there are frequent position changes of MN.

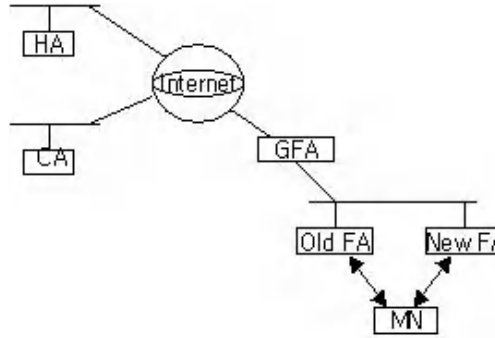


Fig. 3. Network Architecture of MobileIpv6

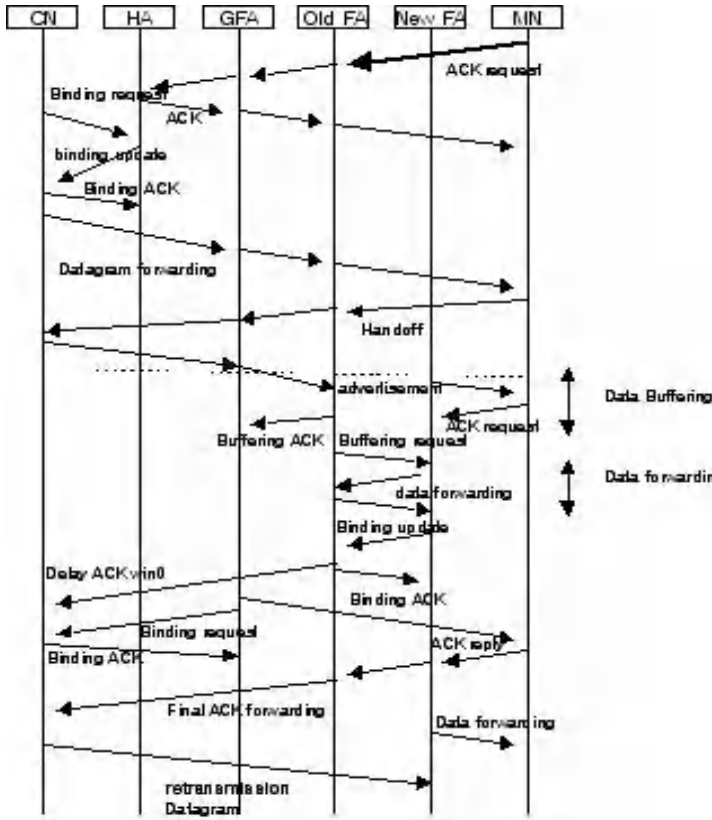


Fig. 4. GFA used handoff protocol

3 GFA Used Handoff Protocol

Whenever MN changes its position in the established mobile IP, it had to execute registration of binding information between HA. However, when there is frequent position change of MN, processing cost for binding increased so the whole wireless network was inefficient in performance.

To improve this defect, a way to reduce the frequent binding update to HA is suggested in IETF by setting up FA based on level and letting an assigned regional GFA monitor position changes of MN within each level.

In case MN moves HA and requires newly updated binding, the data transmission between HA and FA is made via GFA. To say this in another way, if MN changes position to FA binding information that has COA is transferred to HA through GFA. GFA monitors information about characteristics of existing MN in FA that works within its domain and mediates transmitted packet from HA to assigned FA.

Also when MN moves to another FA, binding message of MN is transmitted only to GFA if the movement is made within GFA. In case when this way is used, registration to HA be made only if it is the first binding to MN or it moves to the domain of GFA. Even though MN frequently changes its position, it has less registration to HA, thus, the network processing speed can be improved.

4 Evaluation of Efficiency for GFA Used Handoff Protocol

Based on the introduced handoff protocol above, we used Network Simulator (NS-2.1b7a) to exam the time of transmission delay. The parameter used for the simulation is shown in the table 1.

Table 1. Parameter that is used for simulation

Parameter	Specification
Speed of transmission for wireless section	100kbps
Speed of transmission for wire section	10Mbps
Transmission interval for advertising of agent	1 sec.
Packet size	1kbyte
Overhead for data transmission	60Octats
Overhead for movement	28Octats
Time interval for handoff	5sec.~50sec.
Buffer capacity of agent	1Mbyte

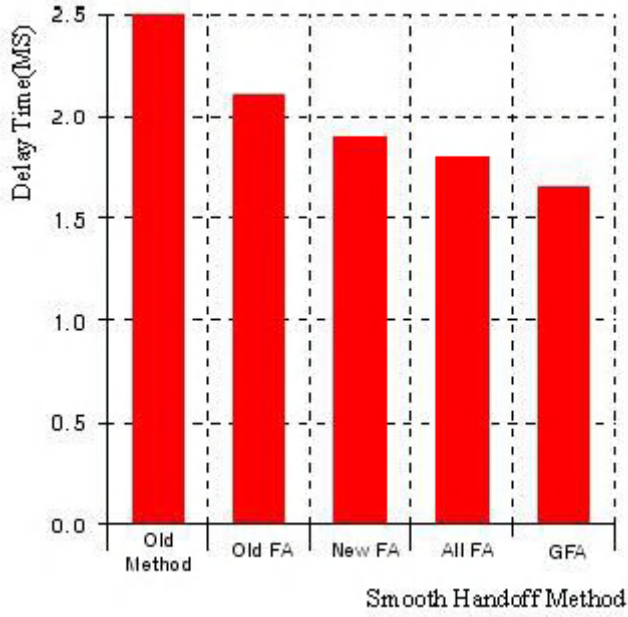


Fig. 5. Handoff Delay Time each Method

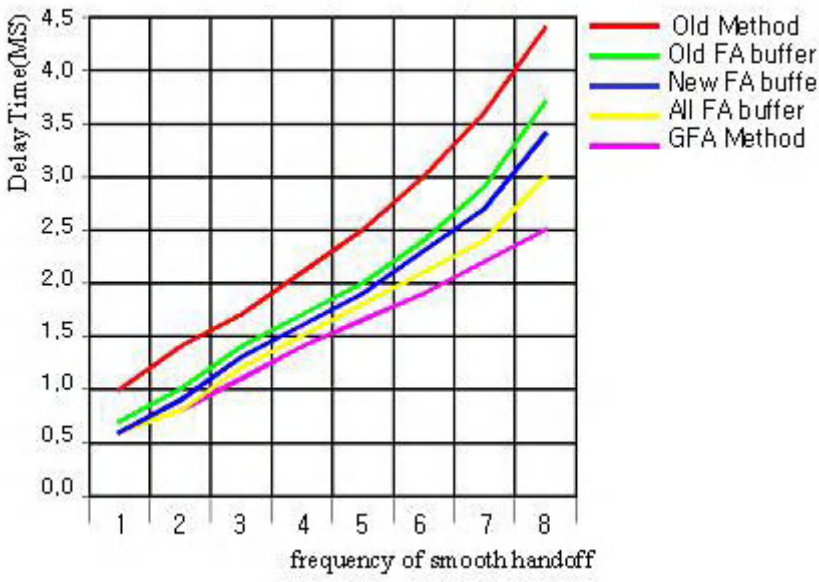


Fig. 6. Frequency of Handoff each Method

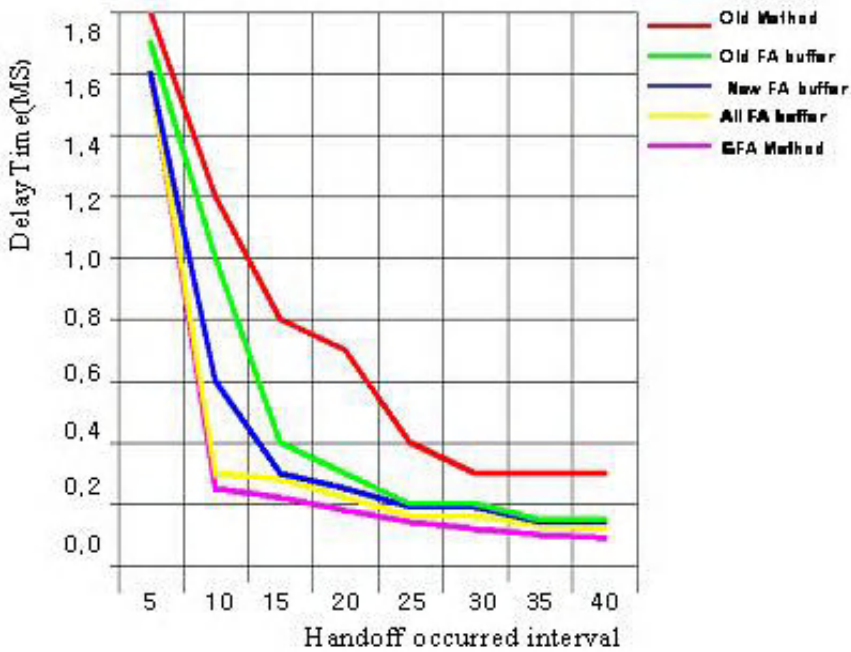


Fig. 7. Handoff occurred interval

The figure 5 compares delay times based on the various Smooth handoff types in case when there are 5 times of handoff. Through this experiment, we confirmed that delay time of handoff can be shortened by buffer used type in FA than not used type. Especially, GFA utilized type has an excellent performance than other types.

In figure 6, comparison is made in the characteristics of each type depends on frequency of smooth handoff. When frequency of handoff is less than 3 times, a type that used buffer in every FA and GFA utilized type do not show much difference. However, as the frequency increases, it is clear that there is a significant effect in reducing time when connecting to HA. The results above suggest it is effective to apply GFA especially in the network where there are frequent handoff.

The figure 7 is a graph that measured transmission delay time depends on handoff occurred interval. As you can see in the graph, utilizing GFA in orange color has less transmission delay time than the type that applied every buffer to FA. It is because it transmits and receives binding information only with adjacent GFA without any information from the updated binding message with adjacent HA in case of handoff.

5 Conclusion

As the usage of internet and mobile phone is in high demand in public nowadays, frequency of handoff is also increased as well. However, the previously introduced smooth handoff types consider network only between two FA so it was difficult to cope with a case when there was handoff to the multi-FA.

In this paper, we ran several experiments and examined a fact that the transaction delay in case of smooth handoff can be minimized by utilizing Gateway FA in mobile network where there is frequent smooth handoff. According to the experiments, transmission delay was less with GFA for cases that occurred more than 3 times handoff compare to the previously introduced types.

However, when handoff was less than 3 times, there was not much difference between using buffer in every FA, and it was costly because it had to set a separate GFA in the network.. Therefore as a result of the study, GFA can be used in restriction only where frequent handoff occurs and transmission delay effect for packet will be maximized in case of using it.

References

1. Charles E. Perkins, Mobile IP-Design Principles and Practices, Addison Wesley, 1998.
2. IETF Network Work Group, "IP Encapsulation within IP," RFC 2003 Oct, 1996.
3. R. Koodli and C. Perkins, "Fast Handovers in Mobile IPv6," draft-koodli-moblieip-fastv6-02.txt, march. 2001.
4. Johnson, David B., Perkins, Charles E., Route Optimezation in Mobile IP, draft-IETF-mobileIP-optim-07.txt, November, 1997.
5. Eva Gustafsson et, al., "Mobile IP Regional Registration," IETF draft, Mar. 2000.
6. Charles Perkins, editor, "IP mobility support", RFC 2002, Oct. 1996.
7. Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, Dec. 1998.
8. Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, Apr. 1996.
9. B. Carpenter, C. Jung,"Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, Mar. 1999
10. Charles Perkins, David B. Johnson, "Mobility Support in IPv6", draft-ietf- mobileip-ipv6 08.txt, Nov. 1998. Work in progress.

A Fault-Tolerant Protocol for Mobile Agent

Guiyue Jin[1], Byoungchul Ahn[2], and Ki Dong Lee[3]

[1]Doctorial student, School of Electrical Eng. & Computer Science,
Yeungnam Univ., 214-1, Dae-Dong, Kyungsan, Kyungbuk, 712-749, Korea
jinguiyue@yumail.ac.kr
[2]Professor, School of Electrical Eng. & Computer Science,
Yeungnam Univ., 214-1, Dae-Dong, Kyungsan, Kyungbuk, 712-749, Korea
ahn@yu.ac.kr
[3]Associate Professor, School of Electrical Eng. & Computer Science,
Yeungnam Univ., 214-1, Dae-Dong, Kyungsan, Kyungbuk, 712-749, Korea
kdrhee@yu.ac.kr

Abstract. Mobile agent technology has been proposed for a variety of applications. Fault-tolerance is fundamental to the further development of mobile agent applications. Fault-tolerance prevents a partial or complete loss of the agent. Simple approaches such as checkpointing are prone to blocking, replications scheme is expensive since it has to maintain multiple replicas. In this paper, a new approach rooted from checkpointing is proposed. This scheme can guarantee to detect and recover most failure scenarios in mobile agent systems, even if there occurs machine failure.

1 Introduction

In recent years, the field of mobile agents has attracted considerable attention. Mobile agent technology has been considered for a variety of application, Such as systems and network management, mobile computing, information retrieval and e-commerce. However, before mobile agent technology can appear at the core of tomorrow's business applications, reliability mechanisms for mobile agent must be established. Among these reliability mechanisms, fault tolerance and transaction support are mechanisms of considerable important and are the subject of this paper.

We begin with the definition of a mobile agent. A mobile agent [1] is a computer program that acts autonomously on behalf of a user and travels through a network of heterogeneous machine. Failures in a mobile agent system may lead to a partial or complete loss of the agent. To achieve fault-tolerance, many fault-tolerant mobile agent approaches are proposed. We first show that a simple checkpointing based [2] execution of an agent, even though it ensures that the agent is not lost, is prone to blocking. Replication [3] prevents blocking, the idea is to use replicas to mask the failures. When one replica is down, it can still use the results from other replicas in order to continue the computation. The advantage of this approach is that the computation will not be blocked when a failure happens. However, this fault-tolerant scheme is expensive since we have to maintain multiple physical replicas for just now logical computation. Since a failure is a rare event, it is not cost-effective to maintain multiple replicas. Moreover, every replica has its own data, and the data in all the

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 993–1001, 2004.
© Springer-Verlag Berlin Heidelberg 2004

replicas must be consistent among themselves. On the other hand, the computation on different replicas may not produce the same and correct result. Thus, it is a tough task in preserving replica consistency, especially when the replicas are widely separated, since the latency of the network will affect the speed of consistency checking as well as preservation.

In this paper, the scheme is rooted from checkpointing and models fault-tolerant mobile agent execution as decision problems when there occurs a failure. We also use the checkpointed data [4] to recover the lost agent. Our approach prevents blocking in the mobile agent execution and ensures the exactly-Once execution property. We validate our approach with the simulation.

The remainder of the paper is structured as follows. In the next section, we will describe our agent execution model. Section 3 describes our proposed scheme. Section 4 discusses the simulation results. Finally, concludes the paper.

2 Agent Execution Model

We assume an asynchronous distributed system, i.e., there are no bounds on transmission delays of messages or on relative process speeds. An example of an asynchronous system is the Internet. Processes communicate via message passing over a fully connected network.

A mobile agent executes on a sequence of machines (also called nodes), where a place (also called landing pad [5] or agency [6]) p_i ($0 <= i <= n$) provides the logical execution environment for the agent. Each place runs a set of services, i.e., a set of operations op_0, op_1, …, that act on the local services, which together compose the state of the place. For simplicity, we say that the agent "accesses the state of the place," although access occurs through a service running on the place. Executing the agent at a place p_i is called a stage S_i of the agent execution. We call the places where the first and last stages of an agent execute (i.e., p_0 and p_n) the agent source and destination, respectively. The sequence of places between the agent source and destination (i.e., p_0, p_1, … p_n) is called the itinerary of a mobile agent. Whereas a static itinerary is entirely defined at the agent source and does not change during the agent execution, a dynamic itinerary is subject to modifications by the agent itself.

Logically, a mobile agent executes in a sequence of stage actions (see the Fig. 1). Each stage action sa_i consists of potentially multiple operations such as, op_0, op_1, and so on. Agent a_i ($0 <= i <= n$) at the corresponding stage S_i represents the agent that has executed the stage actions on places p_j ($j < i$) and is about to execute on place p_i. The execution of a_i on place p_i results in a new internal state of the agent as well as potentially a new state of the place (if the operations of an agent have side effects, i.e., are non-idempotent). We denote the resulting the agent a_{i+1}. Place p_i forwards a_{i+1} to p_{i+1} ($i<n$).

3 Protocol Overview

In this section, we will make statements for system architecture, protocol design and failure and recovery scenarios.

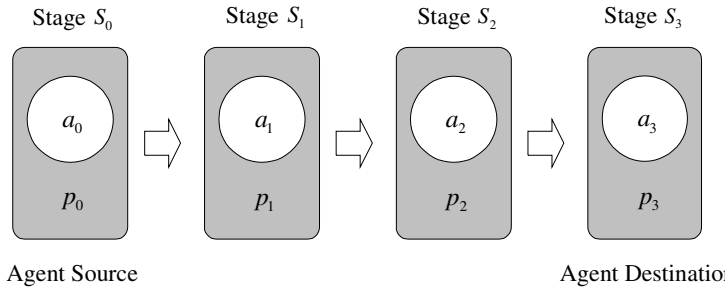


Fig. 1. Model of a mobile agent execution with four stages

3.1 System Architecture

In our agent system, in order to detect the failures of an agent as well as recover the failed agent, our protocol leaves rear guards [7] and the backup of the agent behind whenever an agent moves from one place to another. These rear guards are responsible for monitoring the execution of actual agents and launching a witness agent when there is a failure or they suspect there is a failure. The witness agent is responsible for checking whether the agent is alive or dead. In every place, there is a logging service to write the actions performed by an agent. The information logged by the agent is vital for failure detection. This log file is required when performing the rollback recovery.

We have to design a permanent storage for checkpointing data (the latest agent version) to continue the execution from the checkpoint when the failure occurs. While the actual agent finished the execution, it sends the latest agent version to the rear guards. If there is a failure or the rear guards suspect there is a failure. The lost agent can be recovered by the witness agent using the checkpointing data, or the rear guards send new latest agent to the new place using the checkpointing data, when the failure occurs, the execution can not proceed. If we allow the agent to begin from the staring point of the itinerary, perhaps the exactly-once property will be violated, therefore we have to checkpoint the data of an agent. The overall design of the server architecture is shown in Fig. 2.

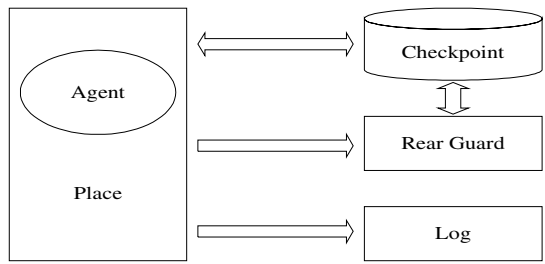


Fig. 2. The server design

3.2 Protocol Design

Our protocol is based on logging and checkpointing as well as decision scheme to achieve failure detection. We discuss the behavior of the agent a_i first. After a_i has arrived at place p_i, it immediately writes a log LOG (i, arrive) on the permanent storage in local log. The purpose of this log is to let the witness agent know that a_i has successfully landed on this place. The agent a_i performs the computations delegated by the owner on Place p_i, and writes logs after every operation. When the actual agent finishes execution, it immediately writes the state and code of the agent on the current place as checkpointing to prevent the loss of the agent. Then it writes a log LOG (i, leave) in the local log. The purpose of this message is to let the witness agent know that a_i has completed its computation, and it is ready to travel to the next place p_{i+1} in the next stage, in the end, a_i sends message MEG (i, leave) the rear guards to inform them that a_i is ready to leave place p_i and sends the latest agent version to the rear guards to recover from the failure when it occurs the failures. In the meantime, a_i leaves place p_i and travels to Place p_{i+1}.

On the other hand, the rear guards listen to the messages coming from the actual agent. The rear guards expect receiving message MEG (i, leave) and the checkpointing data (i.e., latest agent version). When they have received the message and checkpointing data, they do nothing. But if they have not received the message MEG (i, leave), they will suspect that the actual agent fails. After executing decision problem they would send the witness agent to place p_i to deal with the failure. The witness agent can directly communicate with the actual agent with the highest priority for the actual agent. The processing solving the failure will be discussed in detail next section.

The degree of fault tolerance is determined by the number of copies (i.e., checkpointing data) stored on checkpoint in stages where the agent previously has executed. In other words, if the agent a_i is currently executing on place p_i, checkpoints of places p_{i-1},p_{i-2} ,… may store the latest agent copy a_{i-1}. The higher this number, the more concurrent failures can be handled. However, a high number of copies also increases the storage overhead for mobile agent at multiple locations and increase the communication overhead among rear guards when occurring the failure.

3.3 Failure and Recovery Scenarios

In the following subsections, we will cover different kinds of failures including the loss of the agent, the crash of place and crash of machine. While the agent a_i is executing on place p_i at stage S_i, its execution is monitored by rear guards in the previous place. We describe several scenarios as follows

[Case 1] : There are several reasons that an ancestor rear guard fails to receive MEG (i, leave). In this subsection, let's consider the agent's failure. If the rear guards do not receive the MEG (i, leave) after timeout, they would suspect the agent execution is blocked and send a witness agent to place p_i. When the witness agent arrives at the place p_i, first of all, it sends "I arrive" message to the rear guards, and then it checks the log for the agent a_i. If the witness agent can not find the LOG (i, arrive) in the logs, it can be confirmed that the actual agent is lost. The lost agent can by recovered using the checkpointing data taken by the witness agent, so the agent execution proceeds.

[Case 2] : When the rear guards can not receive the MEG (i, leave) , the witness agent is sent to the place p_i , similar to the case in the above subsection, the witness agent sends "I arrive" message to the rear guards first, after the witness agent checks the logs for the agent a_i, the witness agent finds the LOG (i, arrive) in the logs. So the witness agent can confirm the actual agent arrives at the place p_i successfully. Then the witness agent continues checking the log, if the witness agent can not find the LOG (i, leave), the witness agent can confirm the execution has not finished. But it can not know whether the agent execution is still alive or it is dead. we have mentioned that the witness agent can communicate with the actual agent directly, so in this case, the witness agent sends a request "Are you alive ?", if the actual agent responses with "I am alive". The witness agent knows that the agent execution is still active and the witness agent does nothing, otherwise, if the actual agent does not response, the witness agent considers that the agent execution is blocked. Because the agent has only executed partially on this place, it is required to rollback those operations by the method proposed in [8] in order to initialize the execution environment. And then the agent execution proceeds.

[Case 3] : After sending the witness agent to the place p_i, the rear guards do not receive the "I arrive" message and the message MEG (i, leave) also. The rear guards suspect that the machine is down and the agent execution is blocked. The rear guards launch their latest copy of the agent and send it to another place p_i' (see the Fig. 3). Compared with the above two cases, we can see that it is not reliable failure detection, especially in an asynchronous system such as the Internet, no boundaries exist on communication delay or on relative process speeds. Sending the latest copy of agent to place p_i', however, may lead to duplicate agents. Indeed, assume that the rear guards erroneously detection the failure of the agent execution, whereas the execution actually has not failed, in this case, two replicas of agent are executed and it is violated with the exactly-once property. When the rear guards send the latest checkpoint agent to new place p_i', the new actual agent is supposed to new commit decision approaches (i.e., commit-at-destination). That is to say, while the agent has not reached the destination, the local transactions are not committed / aborted yet. We can see that if the machine is not down after rear guards send the agent to new place p_i', the rear guards before the stage S_i receive the message MEG (i, leave) from the original agent. When the new spawn agent arrives at the destination and request the rear guards before the stage S_i, they will deny its commit request. The spawn agent have to abort all execution to guarantee the exactly-once property.

3.4 The Decision Scheme among the Rear Guards

The rear guards are monitoring the agent execution and can resolve the failure when the agent execution is blocked. Because there are several rear guards monitoring the execution, even if there raises a failure in the rear guards, they also can resolve the failure. But among the several rear guards, they have to agree with the decision.

In order to reduce communication overhead, it is not required that the active agent send the message MEG (i, leave) and checkpointing agent data to all the rear guards, but to the latest n rear guards. N is dependent on the failure rate of rear guards and it is enough if it cannot affect the failure detection.

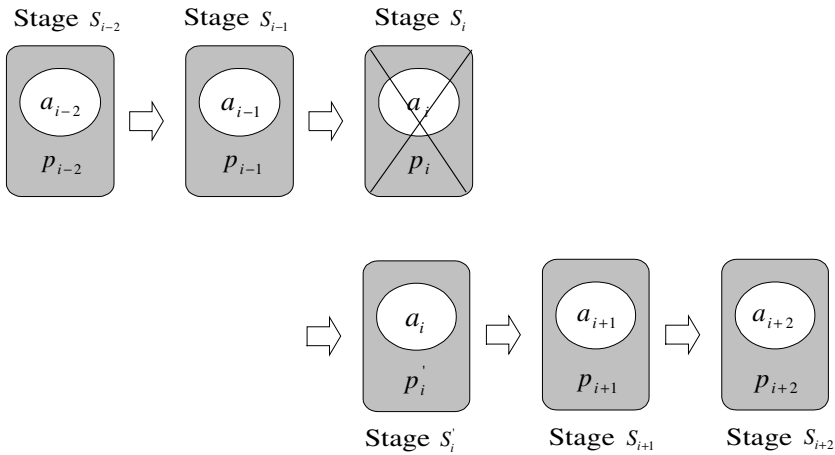


Fig. 3. Upon detection of p_i's failure, p_{i-1} sends a_i to pi'

If all the rear guards receive the message MEG (i, leave) and checkpointing agent data, they do nothing. We can see if there are no failures or very low failure rates, this decision scheme can reduce the communication overhead. If any of the rear guards does not receive the MEG (i, leave), it broadcast to other rear guards the MEG (i, no), if any of other rear guards have received the MEG (i, leave) and checkpointing agent data, it will broadcast the MEG (i, leave) and checkpointing agent data to all other rear guards. For every rear guard, if they receive the message MEG (i, leave) and checkpointing agent data from one of other rear guards, they can confirm the agent execution at stage S_i is completed successfully. If all the rear guards have not received the message MEG(i, leave) and checkpointing agent data, they will elect the latest active rear guard to send a witness agent to the place p_i for solving the failure. Even if the witness agent is sent, all the rear guards have not received any response, they will do the decision process mentioned in the above. And then send the latest checkpointing agent to the new place p_i'. This decision scheme is very effective when the failure rate is very low, but it is not good scheme when the failure rate is high.

4 Simulation and Discussion

To measure the cost of the schemes, we built a simulation system following the mobile agent system environment in a 100 Mbps Ethernet network. When the failure rate is low as shown in Fig.4, the timeout value plays an important role in the replication scheme. For a short timeout period, the frequency of redundant execution may increases, while for a long timeout period, the late failure detection may cause the delay in execution. In the proposed scheme, for a short timeout period, there is no frequency of redundant execution, for a long timeout period, the late failure detection may also cause the delay in execution.

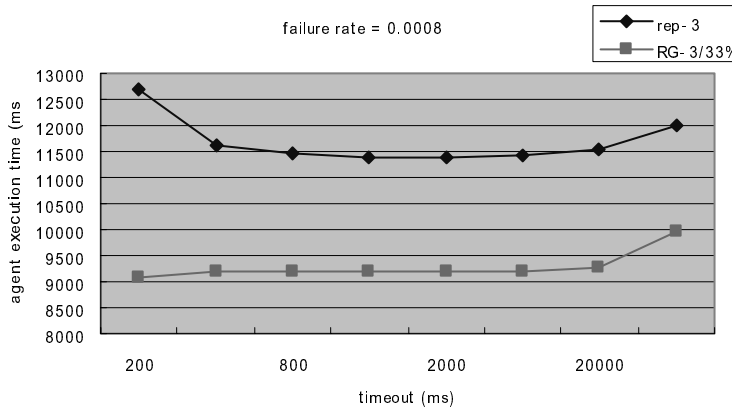


Fig. 4. Agent execution times with various timeout when failure rate is 0.0008

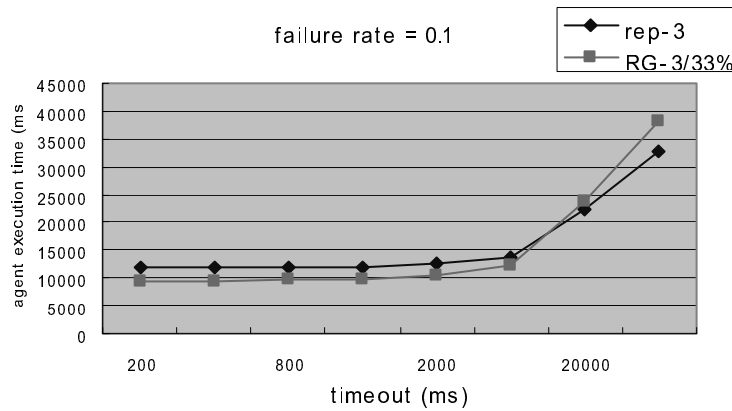


Fig. 5. Agent execution times with various timeout when failure rate is 0.1

When the failure rate is high as shown in Fig. 5, the late failure detection causes the more serious problem. For a short timeout period, there is no influence for agent execution time, while for a long timeout period, the late failure detection causes very serious delay in agent execution.

To measure the cost of the proposed schemes in more various failure environments, the behavior of agents has been simulated with various values for the failure interval. Fig. 6 shows the influence of the failure rate on both of schemes when the timeout is 10000ms. In the replication scheme, because there is little frequency of redundant execution when the timeout is 10000ms, there is no explicit change in agent execution time when the failure rate is low. In the proposed scheme, when the failure rate is low, even if there is 100% of node crash, the performance is not sensitive to the change of the failure rate. When failure rate is high, the performance is sensitive to the change of the failure rate, i.e., the agent execution time for 50% of

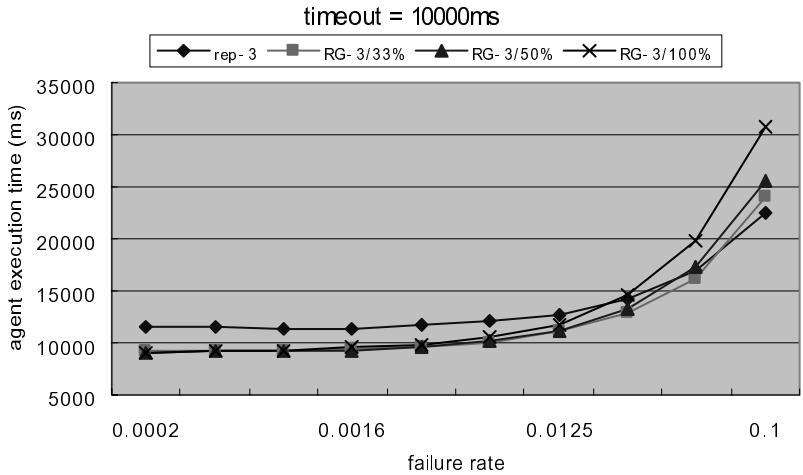


Fig. 6. Agent execution times with various failure rate

node crash is more than that for 33% of node crash and the agent execution time for 100% of node crash is more than that for 50% of node crash. It also shows that when the failure rate is low, the performance of the proposed scheme is better than that of the replication scheme, while the failure rate is high, the performance of the proposed scheme is worse than that of the replication scheme.

5 Conclusion

In this paper, we have identified two important properties for fault-tolerant mobile agent execution: non-blocking and exactly-once and have proposed new fault-tolerant algorithm for mobile agent. The algorithm can ensure that the agent execution proceeds despite a single failure of either agent or place and node (machine) and ensure the exactly-once properties because only one mobile agent runs every step. Our algorithm is rooted from checkpointing scheme and deals with the failure by decision problems among the rear guards. It is simple and effective when the failure rate is low. Our solution can reduce the communication overhead, because it runs the decision scheme only the failure occurs. On the contrary, the replication scheme runs the consensus problem every step. The original checkpointing scheme is prone to block, and although the replication can ensure non-blocking and exactly-once properties, it has lots of communication overhead and it is sensitive to the timeout. From the simulation results, we can see that although the performance of the proposed scheme has a worse performance than that of replication scheme when the failure rate is high, it is not common that the failure rate is too high. In a word, our proposed scheme not only can ensure the non-blocking and exactly-once properties, but also has a better performance than the replication scheme.

References

1. The Object Management Group: The Mobile Agent System Interoperability Facility, OMG TC Document orbos, URL: http://www.omg.org. (2000)
2. Eugene, G., Lubomir, F. B., and Michael, B. D.: An Application-Transparent, Platform-Independent Approach to Rollback-Recovery for Mobile-Agent Systems, Proc. of 20 th IEEE International Conference on Distributed Computing Systems, Taiwan, (2000)
3. Stefan, P. and Andre, S.: FATOMAS- A Fault-Tolerant Mobile Agent System Based on the Agent-Dependent Approach, Proc. of the international Conference on Dependable Systems and Networks, (2001) 215–224
4. Victor, F. N.: Checkpointing and the Modeling of Program Execution Time, in Software Fault Tolerance, Michael R. Lyu(.ed) John Wiley & Sons, (1994) 213–248
5. Dag, J., Keith, M., Fred, B. S., Kjetil, J. and Dmitrii, Z.: NAP: Practical Fault-Tolerance for Itinerant Computations, Proc. of the 19 th IEEE International Conference on Distributed Computing Systems, (1999) 180–189
6. Luis, M. S., Vitor, B. and Joao, G. S.: Fault-Tolerant Execution of Mobile Agents, Proc. of the International conference on Dependable systems and Networks(DSN 2000), (2000) 135–143
7. Dag, J., Robbert, R. and Fred, B. S.: Operating System Support for Mobile Agents, Proc. of the 5 th Workshop on Hot Topics in Operating Systems, Washington, (1995) 42–45
8. Philip, A.B., Vassos, H. and Nathan, G.: Concurrency Control and Recovery in Database Systems, Addison Wesley, Reading, Mass., (1987)

Performance Analysis of Multimedia Data Transmission with PDA over an Infrastructure Network*

Hye-Sun Hur[1] and Youn-Sik Hong[2]

[1] Department of Information and Telecommunication Engineering
[2] Department of Computer Science and Engineering
University of Incheon, 177 Dohwa-dong Nam-gu, Incheon 402-749, Korea
{mshush,yshong}@incheon.ac.kr

Abstract. Various experiments were conducted to measure and analyze the performance of transmission of multimedia data with PDA as a mobile host over an infrastructure network. Our test-bed network consists of the infrastructure network integrating a wireless network based on IEEE 802.11b with wired Internet. For an analysis of the performance, we have measured the time of transferring a wave file (about 469Kbytes) from PDA to a desktop server or vice versa over the wireless network by changing the size of a packet from 256 bytes up to 7128 bytes on application layer. As the size of a packet becomes larger, the time elapsed to transfer the data from PDA to the server or vice versa becomes shorter in the wireless network. Even with the size of a packet greater than the maximum segment size of TCP on transport layer, the transmission time also becomes shorter inversely proportional to the size of a packet.

1 Introduction

An Internet service based on 802.11b standard wireless LAN with 11Mbps transmission speed has been carried out since the wireless LAN service using 2.4GHz ISM(*Industrial, Scientific and Medical*) band has recently been permitted. It is expected that applications that can be maximally utilized with multimedia data will be a main stream in near future in case that 802.11a wireless LAN service using 5GHz UNII(*Unlicensed National Information Infrastructure*) band is feasible.

Currently, we have interested in transmission of multimedia data over a wireless network when PDA(*Personal Digital Assistant*) is used as a mobile host(MH) instead of laptop PC(hereinafter *laptop*). Much of the works have been focused on the performance analysis of data transmission over a wireless network using laptop as MH. We think that PDA has advantage over laptop in the point of mobility due to its smaller size. But, PDA is inferior to laptop in the overall performance. Thus, in this paper we are focusing on the performance analysis of data transmission over a wireless network when we are using PDA as MH.

Another consideration is the packet size to be transferred. Most of the works have restricted their packet size to 1460 bytes, which is the maximum segment size(MSS)

* This work was partially supported by the Korea Science and Engineering Foundation (KOSEF) through the Multimedia Research Center at University of Incheon.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 1002–1009, 2004.
© Springer-Verlag Berlin Heidelberg 2004

of TCP on transport layer. In this paper, to take an effect of packet size on the performance, we have used the packet size beginning from 256 bytes up to 7128 bytes on application layer.

A bulk data has been taken as the sample multimedia data for our experiments. We created a wave file(469 Kbytes) by recording one's voice for 1 minute with 8000Hz mono sampling rate.

This paper consists of the following; we discuss about the previous works in chapter 2. In chapter 3, we briefly explain about Voice Messenger System that is implemented for that the all experiments are made. In chapter 4, the performance metrics have been explained. Chapter 5 presents the experimental results conducted under various conditions and the analysis are dealt with. Finally, we conclude our works in chapter 6.

2 Research Scope and Related

An infrastructure network is a network combining BSS(*Basic Service Set*) which is a set of wireless terminals and a AP with wireless LAN based on Ethernet. Several BSS can be integrated under this network environment[1]. We have constructed the infrastructure network as shown in Fig. 1 integrating wired Internet based on Ethernet and a wireless network based on IEEE 802.11b standard. The wireless network in Fig.1 consists of one BSS.

TCP(*Transmission Control Protocol*) may not be suitable for a wireless network with high packet error rates(PER) because TCP is originally designed for a wired network with low PER. But *Balakrishnan* et. al.[2] insisted that adapting TCP to a wireless network would be suitable for achieving reliability and they suggested Snoop protocol which is a modified version of TCP. In this paper, we carried out experiments using general TCP.

Chang et. al.[3] calculated PER by adjusting the packet size between 100 and 400 bytes, on the assumption that 802.11b wireless LAN of 2.4GHz ISM bandwidth can make packet losses due to the interference of a microwave oven and a cordless phone using the same bandwidth. They proved that the bigger the packet is, the more interference of wireless frequency it gets.

The wireless mobile terminals used in the relevant studies were laptops[2]. As explained earlier PDA is used as MH in our experiments.

3 Voice Messenger Systems Works

We have designed and implemented VMS(*Voice Messenger Systems*) as shown in Fig. 1 in order to measure the performance of multimedia data transmission under the infrastructure network. VMS is a kind of file transfer system, which consists of VMS server as FH(*Fixed Host*) and clients as MH. Each VMS client records one's voice and sends it to the VMS server after converting it into a wave file. VMS server receives this voice message sent from its client and stores it in its hard disk. The server transfers this message to the authenticated client that has needed the stored message.

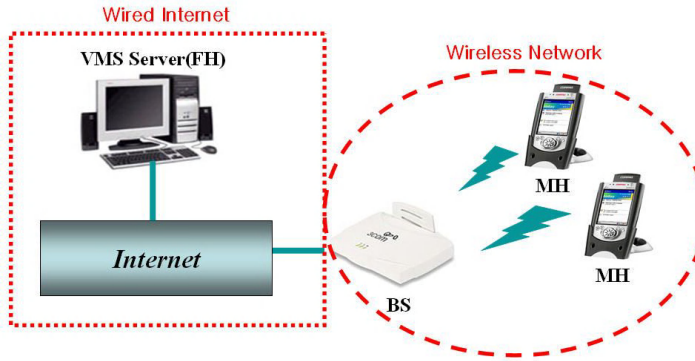


Fig. 1. The infrastructure network

Table 1. The specification of VMS components

VMS components		*Processor*	*RAM*	*NIC*
MH	PDA	Strong Arm 206MHz	64MB	1)PCMCIA, 11Mbps 2)CF, 11Mbps
FH	Desktop PC	Pentium 2.4GHz	512MB	PCI, 100Mbps

In Fig.1, PDA is used as MH and PC is used as FH. The detailed specification of VMS components are shown in Table 1.

AP is used as BS(*Base Station*) and its maximum transmission speed is 11Mbps. NIC(*Network Interface Card*) and AP are needed to connect a wired and a wireless network. In this paper, two different types of wireless NIC, PCMCIA and CF(*Compact Flash*), are used for MH NIC. MH NIC is a wireless NIC designed according to IEEE 802.11b standard with the transmission speed of 11Mbps. The transmission speed of NIC of the VMS server is 100Mbps.

4 Performance Metrics

We start by defining the terminologies. *Transmission time* means the time elapsed to send/receive the message(data) completely between the VMS server and its client. We call the time of receiving messages for its client from the VMS server *receiving time* and the time of sending messages to the VMS server *sending time.*

Then we assume the following conditions for our experimentation.

- The data to be sent is the wave file of recording one's voice for 1 minute, where its file size is 469Kbytes(480,046 bytes in exact size). PDA as VMS client can either send or receive such a wave file to/from the VMS Server.
- The maximum transmission unit(MTU) of TCP is 1500 bytes in both wired and wireless network. Thus, the maximum segment size(MSS) of the TCP segment to be used during data transmission is 1460 bytes[4].

A wireless network basically uses packet switching of data transmission in a unit of packet size. In application layer, a larger file should be divided into smaller packet units and then transfer them on transport layer.

The MSS of the TCP segment is 1460 byte which can pass through a router without fragmentation. In the experiments suggested in various papers including *Chang* et. al.[3] they have used the size of the packet less than 1460 bytes[5]. Besides, in the case of mobile phone it is generally to set a size of its packet to 512 bytes when it is used for transferring multimedia data over the air.

The size of a packet is expected to grow rapidly because 802.11a using 5GHz UNII band can increase the maximum speed of data transmission up to 54Mbps even for a wireless network. Therefore we set the size of a packet greater than 1460 bytes on application layer for the experiment in this paper.

- We designated 14 different sizes of the packets to be sent from 256 bytes to 7128 bytes.

For our experiments, we have made the following 3 scenarios:

Scenario 1. Compare the transmission times of two different types of wireless NIC(CF and PCMCIA) in a wireless network to check which one is better under the same condition.

Scenario 2. Measure the RTT(*Round Trip Time*) and the loss of packets depending on the size of the packets to be sent in a wireless network. To do this, we use PING program supported by Linux systems. With the same metrics, we will do the same experiments for a wired network. By doing so, we will try to find out the difference between the wireless network and the wired network based on such metrics.

Scenario 3. Measure and compare the transmission time by increasing the packet size starting from 256 bytes up to 7128 bytes on application layer, not on transport layer. In general, the transmission time may decrease as the packet size increases. Our main concern is to see whether the transmission time may increase or not when the packet size is greater than the MSS of the TCP segment. By doing so, we will try to find the packet size on application layer that gives the best performance of transmission time.

We locate the MH client within 5-meter radius of AP(called *good position*) in all the scenarios above. To do so, radio frequency either radiated by AP or received by wireless LAN could maintain both good signal quality and good signal strength, while keeping away from multi-path and fading effect.

5 Result and Analysis of the Experiment

5.1 Scenario 1

There are two types of wireless NIC; PCMCIA and CF. We used 2 PDA clients; The one has NIC of PCMCIA type and the other has NIC of CF type built in them. Table 2 shows the results of measurement of the transmission time of either sending or receiving the wave file when the two PDA clients are connected to the VMS server at the same time, where the size of a packet to be sent is 4096 bytes.

Table 2. The transmission time of CF and PCMCIA type(unit: ms)

	NIC type	*PDA→Server*	*Server→PDA*
1	CF	3106	13240
2	PCMCIA	2098	12218

The transmission time of PDA client with PCMCIA NIC built in it is relatively constant. However, the sending time of PDA client with CF NIC(PDA→Server) can be increased by 1008 ms compared to that of PCMCIA type. We thought a wireless NIC of PCMCIA type would be more stable in a wireless network and decided to use PCMCIA type only from the next experiment.

5.2 Scenario 2

PING program running on Linux systems gives us RTT which allows to predict the transmission delay and PER of a given network. For this experiment, one PDA with the wireless NIC of PCMCIA type as MH and the server running on Linux systems connected to wired Internet have been used. We specify the destination address of PING program as IP address of the PDA client. The 14 different sizes of packet(256 to 7128 bytes) have been applied. For the each packet size the average of 200 repetitions has been calculated(Fig. 2).

With the same metrics used as the wireless network, we did the same experiments for a wired network. We also specify the IP address of FH PC as a destination address to measure RTT and PER of a given wired network(Fig. 3).

We can say that the mean RTT is proportional to the size of packet to be sent both the wireless network and the wired network. As shown in Fig. 3, the mean RTT of the wired network is the minimum of 0.22 ms when the size of a packet is 256 bytes. It also gives the maximum of 1.61 ms when the size of a packet is 7128 bytes.

The mean RTT in a wireless network increases as the size of a packet becomes larger as happened in wired network. As shown in Fig. 2, the mean RTT is 4 ms when the size of a packet is 256 bytes and 31 ms when it is 7128 bytes. The mean RTT of the wireless network is 18 times longer at maximum than that of a wired network.

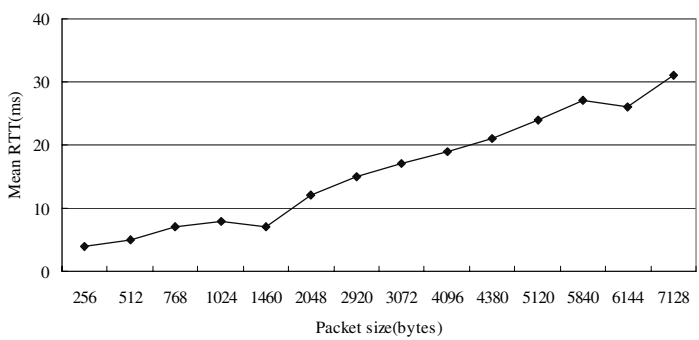


Fig. 2. The mean RTT over a wireless network

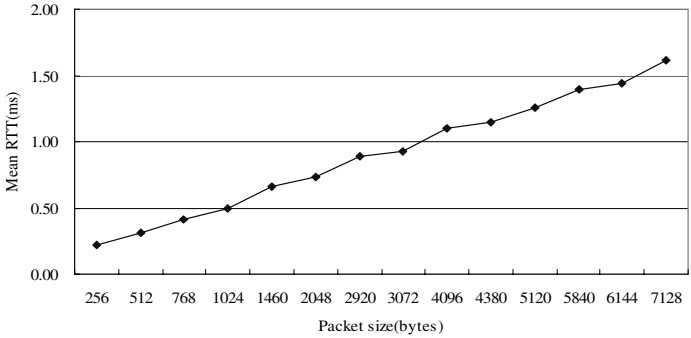


Fig. 3. The mean RTT in a wired network

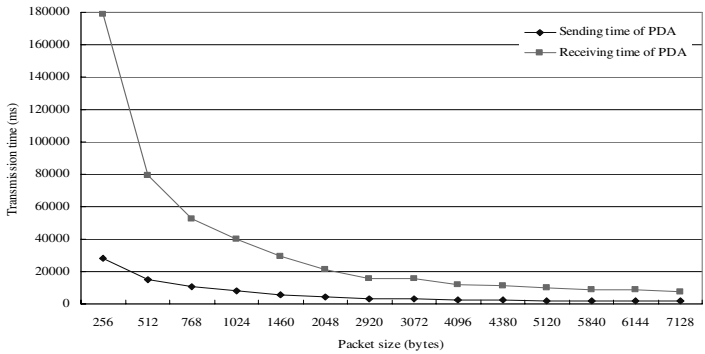


Fig. 4. The transmission time measured for the size of a packet

In another words, in a wired network a network delay does not happen even when the size of a packet becomes larger. However, in a wireless network the occurrence of packet retransmission may happen because the signal strength gets weak due to the multi-path and fading effects. Thus, the bigger the size of a packet divided randomly to transfer data in a wireless network, the more delays in the network there will be.

However the occurrence of packet loss is hardly generated in a wired network and a wireless network either. We come to a conclusion that the size of a packet has almost nothing to do with the PER in a wireless network. *Nguyen* et. al.[6] presented the similar results as ours that the PER is less than 0.001 within 5m radius of AP.

5.3 Scenario 3

We have measured the time of transmission of the wave file(469Kbytes) from one PDA client to the VMS server or vice versa with each of the 14 different packet sizes. In order to analyze the results of the Scenario 3, Analyzer 2.2[7] which is a kind of network monitoring tool has been used.

Fig. 4 shows the transmission time measured depending on the size of a packet. The experimental results say that when the size of a packet becomes larger, its transmission time becomes shorter. In other words, when the file to be transferred is

Table 3. When PDA client sends data; The number of packets and ACKs and their average time for the selected size of packet

Packet size (bytes)	No of packets	Average transmission time(ms)	No of ACKs	Average response time(ms)
1460	329	17.985	166	0.028
2920	329	10.023	166	0.042
4380	329	6.947	166	0.035
5840	329	4.534	167	0.027

divided by a larger size of a packet, the total number of packets to be sent gets smaller and thus its transmission time becomes shorter.

In addition, as shown in Fig. 4, it is clear that the time of receiving the data for PDA client from the server takes longer by 4.2 times at the minimum (with the packet size of 7128 bytes) to 6.3 times at the maximum (with the packet size of 256 bytes) than the time of sending the data from PDA client to the server. The reason is that the window size of PDA client and that of the server are quite different. From the outputs of the packet monitoring using Analyzer 2.2, the window size of PDA client is 17,520 bytes, while the window size of the server is 32,768 bytes. Thus in the case of sending data for PDA client to the server, it takes shorter because its window size is larger compared to that of PDA client. Conversely, when PDA client is received data from the server, it takes longer because its window size is smaller than that of the server.

In Table 3, we summarized the number of packets and the number of ACKs (acknowledgement) and their average time for the selected size of packet which is multiple of the MSS, that is 1460 bytes, of TCP, when PDA client sends data to the server.

Notice that the number of packets and the number of ACKs are almost equal for the 4 different packet sizes. However, as the size of packet becomes increases, the average transmission time gets shorter. Even though the average response time for the ACK varies with the size of packet, their effect on the overall performance is relatively low. Therefore, we can say that in the wireless network, the overall performance can be improved in proportional to the size of a packet. Furthermore, with the size of a packet greater than the MSS of TCP, the similar result as we have expected has been achieved.

6 Conclusions

Various experiments were conducted in order to measure and analyze the performance of transmission of multimedia data with PDA as a mobile host over an infrastructure network. Since the amount of data to be transmitted on transport layer is restricted to the MSS of TCP, 1460 bytes, the previous works limited their packet size to it. However, in our experiments, even with the size of a packet greater than the MSS of TCP on application layer, the overall performance is proportional to its size. In other words, as the size of a packet becomes larger, the transmission time gets shorter. Besides, the window size of PDA as a mobile host is smaller by half than that of the server as a fixed host in our infrastructure network. Thus, the time of receiving data for PDA from the server takes longer by 4-6 times than the time of sending data from PDA client to the server.

References

1. IEEE 802.11 Standard. http://standards.ieee.org/getieee802/802.11.html (1999)
2. Balakrishnan, H., Seshan, S., Amir, E., Katz, R. H.: Improving TCP/IP Performance over Wireless Networks. MOBICOM 1995 (1995)
3. Chang, W-C., Lee, Y-H., Ko, C-H., Chen, C-K.: A Novel Prediction Tool for Indoor Wireless LAN under the Microwave Oven Interference. ISW 2000 (2000)
4. Stevens, W. R.: TCP/IP Illustrated Volume 1: The Protocols. Addison-Wesley (1994)
5. Balshi, B. S., Krishna, P., Vaidya, N. H., Pradhan, D. K.: Improving Performance of TCP over Wireless Networks. ICDCS 1997 (1997)
6. Nguyen, G. T., Katz, R. H., Noble, B., Satyanarayanan, M.: A Trace-Based Approach for Modeling Wireless Channel Behavior. Winter Simulation Conference (1996) 597–604
7. Analyzer 2.2, http://analyzer.polito.it

A New Synchronization Protocol for Authentication in Wireless LAN Environment*

Hea Suk Jo and Hee Yong Youn

School of Information and Communications Engineering
Sungkyunkwan University, 440-746, Suwon, Korea +82-31-290-7952
jojo@skku.edu, youn@ece.skku.ac.kr

Abstract. Today, wireless LANs are widely deployed in various places such as corporate office conference rooms, industrial warehouses, Internet-ready classrooms, etc. However, new concerns have been raised regarding security. Currently, both virtual private network (VPN) and WEP are used together as a strong authentication mechanism. In this paper a new synchronization protocol for authentication is proposed which allows simple authentication, minimal power consumption at the mobile station, and high utilization of authentication stream. This is achieved by using one bit per a frame authentication, while main authentication process including synchronization is handled by access points. Computer simulation reveals that the proposed scheme significantly improves the authentication efficiency in terms of the number of authenticated frames and authentication speed compared with earlier protocol employing a similar authentication approach.

1 Introduction

Wireless communications offer organizations and users many benefits such as portability and flexibility, increased productivity, and lower installation costs. Wireless technologies cover a broad range of differing capabilities oriented toward different uses and needs. Wireless local area network (WLAN) devices, for instance, allow users to move their laptops from place to place within their offices without the need for wires and losing network connectivity. Less wiring means greater flexibility, increased efficiency, and reduced wiring costs. Organizations are rapidly deploying wireless infrastructures based on the IEEE 802.11 standard [3].

Unfortunately, the 802.11 standard provides only limited support for confidentiality through the wired equivalent privacy (WEP) protocol which contains some flaws in the design [5]. Therefore, users should be aware of the security risks associated with wireless technologies, and need to develop strategies that will mitigate the risks as they integrate wireless technologies into their computing environments [4].

* This work was supported in part by 21C Frontier Ubiquitous Computing and Networking, Korea Research Foundation Grant(KRF – 2003 – 041 – D20421) and the Brain Korea 21 Project in 2003. Corresponding author: Hee Yong Youn.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 1010–1019, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Currently, both virtual private network (VPN) and WEP are used together as a strong authentication mechanism. With IPsec/VPN, security services are provided at the network layer of the protocol stack. This means all applications and protocols operating above the network layer are IPsec protected. The IPsec security services are independent of the security at layer-2, the WEP security. As a defense-in-depth strategy, if a VPN is in place, an agency can consider having both IPsec and WEP. Like WEP, VPN supports strong authentication between mobile station (STA) and access point (AP). Authentication is a serious issue in wireless environment where the bandwidth resource and power are severely limited. This paper thus proposes an efficient authentication protocol, which allows authentication with significantly lower overhead than earlier approaches.

In the literature there exist several protocols solving the problem of authentication in wireless security. Recently, [1,2] proposed SOLA (Statistical One-bit Lightweight Authentication) approach where one bit is used for a frame authentication. In this paper a new frame synchronization protocol is proposed using a similar approach as SOLA but much more efficient than that. The main features in the design of the new synchronization protocol are the followings:

- *Strong authentication*: It can detect an attack (Denial-of-Service attack, overwrite attack, Man-in-Middle attack) with high probability.
- *Simple authentication*: Authentication using only one bit for resource-constrained environments like wireless networks.
- *Minimal power consumption in the mobile station*: The mobile station is responsible for synchronization of the authentication stream in SOLA. In the proposed protocol access point is responsible for synchronization, resulting in less power consumption in the mobile station.
- *High throughput*: The number of authenticated frames is larger than SOLA for a given length of authentication stream. Computer simulation reveals that the proposed scheme substantially increases the number of authenticated frames compared with SOLA protocol in practical operational environment. This eventually in creases the speed of authentication.

The remainder of the paper is organized as follows. Section 2 reviews the 802.11 access control mechanisms and related work. Section 3 presents the proposed authentication protocol, and Section 4 evaluates it. Section 5 concludes the paper.

2 Review of IEEE 802.11 Access Control Mechanisms

The 802.11 wireless LAN standard incorporates three mechanisms to provide secure client access to wireless LAN access points, including the Service Set Identifier (SSID), Media Access Control (MAC) address filtering, and Wired Equivalent Privacy (WEP) and VPN connections.

2.1 IEEE 802.11 Access Control Mechanisms

802.11 provides some basic security mechanisms to make the enhanced wireless freedom less of a potential threat. The MAC layer supports authentication and privacy

through encryption. In addition, all Wi-Fi access points and end user devices can be configured with a Basic Service Set Identifier (SSID) [9]. This SSID must be known by the Network Interface Controller (NIC) in order to associate with the AP and thus proceed with data transmission on the network. If the SSID does not match the one stored in the AP, then the STA cannot establish a connection to the Wireless LAN. By default, the SSID is not really a wireless LAN security feature but an easy authentication tool because:

- it is well known to all NICs and APs
- it can be controlled by the NIC/Driver locally whether the association is allowed or not if the SSID is not known
- there is no encryption provided through this scheme.

Most APs offer a feature that defines which clients may connect determined by their MAC addresses. A MAC address (media access layer) is a hard-coded address on a network interface card that is different from an IP address. A MAC address is usually static and never changes even when the card is removed from the device. With MAC address filtering turned on, a workstation will not be able to connect unless its MAC address has been defined on the AP. This security feature is useful in small networks, although keeping a list of updated MAC addresses for a large network can be too difficult to manage. Although the list of accepted MAC addresses is difficult, if not impossible, to extract from most APs, it is possible for someone with right tools and knowledge to discover one of the MAC addresses already in use on a network. An attacker could then configure a workstation to masquerade as a legitimate workstation with the stolen MAC address.

The IEEE 802.11b standard stipulates an optional encryption scheme called WEP that offers a mechanism securing Wireless LAN data streams. The goal of WEP is to provide an equivalent level of privacy as is ordinarily present with an unsecured wired LAN. Wired LANs such as IEEE 802.3 (Ethernet) do not incorporate encryption at the Physical or Media Access layer, since they are ordinarily protected by physical security mechanisms such as controlled entrances to a building. Wireless LANs are not necessarily protected by this physical security because the radio waves may penetrate the exterior walls of a building. In the IEEE 802.11 specification process it was decided to incorporate WEP into the standard to provide an equivalent level of privacy as the wired LAN by encrypting the transmitted data.

A Virtual Private Network (VPN) is a way to use a public telecommunication infrastructure such as the Internet to provide remote offices or individual users with secure access to the network of their organization. A VPN can be contrasted with an expensive system of owned or leased lines that can only be used by one organization. The goal of a VPN is to provide the organization with the same capabilities, but at a much lower cost.

2.2 Authentication Protocol

As mentioned earlier, VPNs are used to provide protection for areas where current 802.11 solutions are not enough. Unless some strong costly authentication mechanisms such as IPSec AH/ESP, WEP or AES+OCB in IEEE 802.11 Task Group I are used to protect the data packets, we have no assurance whether some malicious neighbors are impersonating a non-malicious user [2] or not.

Some identity authentication protocols were proposed to detect unauthorized access in 802.11. SOLA (Statistical One-bit Lightweight Authentication) is the most recent protocol which allows a robust lightweight one-bit identity authentication protocol without the need of expensive authentication mechanism. Here an identical random authentication bit stream is generated in both the STA and AP, and one bit in the stream is attached to the MAC-layer header of each data frame sent by the STA for identity authentication at the AP by bit matching. Note that a data frame can be lost easily due to harsh wireless network environment. As a result, synchronization between STA and AP is critical for the approach to be effective. We next present the proposed authentication protocol.

3 The Proposed Protocol

The proposed protocol is for allowing a strong one-bit authentication without any redundancy between WEP and VPN. It is designed to provide efficient identity authentication at the link layer for wireless networks.

3.1 The Overview

WEP security protocol has been implemented for client/server wireless LAN along with APs. However, the wireless network is insecure if only the WEP is used for security. As a result, most companies use both VPN and WEP to secure their networks [13], and it is most common that an IPSec/VPN tunnel is used without any encryption or authentication at the link layer. Therefore, a new option is required at the link layer for per packet identity authentication in access control.

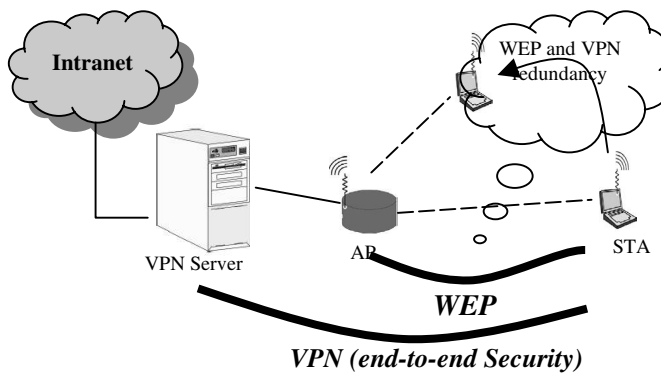


Fig. 1. WEP and IPSec/VPN solution in an 802.11 network.

The main idea of the authentication is that an identical random identity authentication stream is generated in both the STA and AP, and then each successive bit obtained from this stream is attached to the MAC-layer header of each data frame

for identity authentication [2]. As shown in Fig. 1, strong authentication is provided once between the VPN Server and STA, and second between AP and STA if this approach is offered.

The authentication protocol identifies and authenticates an STA, and validates the service request type to ensure that the user is authorized to use particular network services. Specifically, authentication is performed using the bit attached to each data frame. The following explains the basic operation flow.

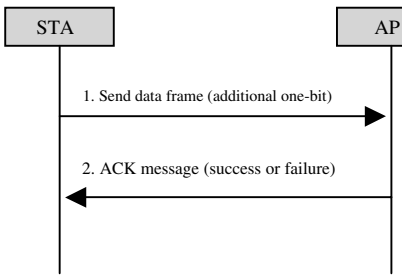


Fig. 2. Successful authentication

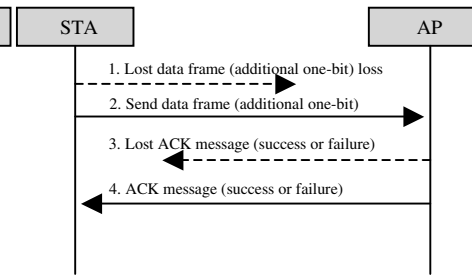


Fig. 3. Authentication with an attack or network obstacle.

Each STA and AP initializes a random bit stream called authentication stream, which is created using a same seed value. The STA and AP receive the seed value of authentication stream at the connection setup. The signaling flows are as follows.

- When an STA successfully sends a data frame to the AP (see Fig 2.)

Step 1. One-bit from the authentication stream is attached to the data frame and then sent to the AP. Unfortunately, data can be lost due to an attack or unexpected network obstacle.

Step 2. When the AP receives a data frame, it compares the bit attached by the STA with the bit generated by itself. If they match according to the synchronization algorithm, then the authentication is successful and the AP sends 'ACK-success' message to the STA. Otherwise, sends 'ACK-failed' message.

- When a data frame or ACK message is lost (see Fig 3.)

Step 1. Data frame is lost due to an attack or unexpected network obstacle. Then the AP will not receive the data frame, and the STA waits for ACK message from the AP.

Step 2. If an ACK message does not arrive for some predefined time limit, the STA sends the same data frame again.

Step 3. The AP receives a data frame, and it sends an ACK message to the STA but the message may be lost due to some problem. As in Step 2, the STA sends the data frame again.

Step 4. According to the condition of match of the authentication bits, 'ACK-success' or 'ACK-failed' message is sent.

3.2 Synchronization

A synchronization algorithm is used to match the bits obtained from both the authentication streams. It is based on the moving pointers of STA and AP authentication stream. If the AP finds the authentication bits are same, both the STA and AP authentication pointer move forward one bit position. Also, the AP sends an 'ACK-success' message to the STA. Otherwise, the AP authentication pointer moves backward one bit position while sending an 'ACK-failed' message.

If the STA receives an 'ACK-success' message from the AP, the STA authentication pointer moves forward one bit. If it receives an 'ACK-failed' message or no message in a time limit, it sends the same data frame again. With the proposed approach, the value of the STA authentication stream pointer does not become greater than that of the AP authentication stream pointer. The frequency of the STA pointer movement is also much smaller than that of the AP pointer, which results in minimal power consumption at the STA. This is a crucial factor for mobile stations of limited power. The synchronization algorithm executed in the AP and STA can be described with the following pseudo code:

```
Algorithm for AP
//AP receives data packet with Bit[a]
    if Bit[a] == Bit[b] then
       b++;
       AP->STA : Packet{ACK, success}
    Else if Bit[a] ≠ Bit[b] then
       b--;
       AP->STA : Packet{ACK, failed}
Algorithm for STA
//STA receives ACK packet with success or failed bit
from STA
    if bit == success then
       a++;
End of Algorithm
```

The following analyzes the synchronization algorithm.

Lemma 1. When the STA and AP are synchronized, the STA's authentication stream pointer(Psta) is always smaller or equal to the AP's pointer(Pap).

Proof : The STA increases the pointer when it receives 'ACK-success' as shown in the Case i) below. When the AP sends 'ACK-failed' to the STA, its authentication pointer is decremented as Case ii). When the 'ACK-success' from the AP is lost, Pap is still incremented since the AP does not know the condition as Case iii). As a result, the AP's pointer value is always greater than or equal to the STA's pointer value.

```
i)   ACK-success  : Psta++, Pap++  so,  Psta = Pap
ii)  ACK-failed   : Pap--  so,  Psta = Pap
iii) AP data loss : Pap++  so,  Psta < Pap
```

As shown in Fig 4., assume that both the STA authentication stream pointer and AP pointer point bit no. 2(①). The STA sends a data frame to the AP, and the STA's pointer is still in the same place. The AP compares the received authentication bit

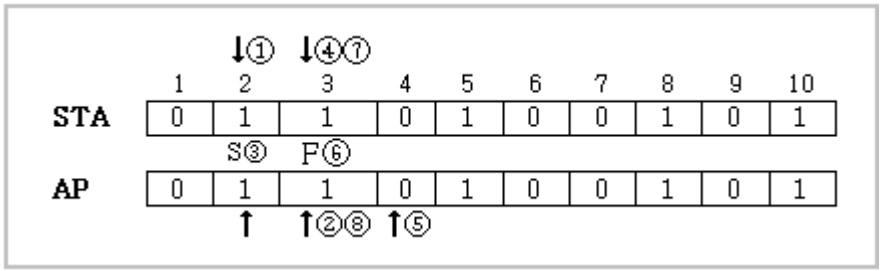


Fig. 4. An example of authentication operation.

with its own authentication bit. It moves the pointer(②) and sends 'ACK-success'(S③) to the STA when the compared values are same. The STA moves its pointer when an 'ACK-success' arrives. By the same way, the STA's authentication bit is sent to the AP(④). The AP moves the pointer if the bits are same(⑤). If the 'ACK message' is lost, the STA sends the bit again(⑦). When the AP compares the bits (1 from STA bit-3 and 0 from AP bit-4) it finds they mismatch and thus decrements its pointer(⑧). As we see in this scenario, the AP's pointer value is smaller or same as the STA's pointer value.

Lemma 2. When synchronization fails, the STA and AP do not know the fact until the authentication bit values mismatch.

```
i)   Psta = Pap ,   *Psta = *Pap
ii)  Psta ≠ Pap ,   *Psta = *Pap
iii) Psta ≠ Pap ,   *Psta ≠ *Pap
```

Proof : In normal case of Case i) the pointer addresses and authentication values are same. In Case ii) the pointer addresses are different but the authentication bit values are same. In this case transmission continues without knowing the streams are unsynchronized. In Case iii) the pointer addresses and authentication bit values do not match.

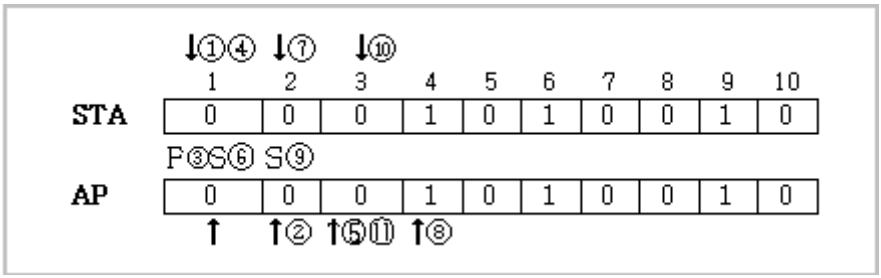


Fig. 5. An example of the case of non-synchronization.

When non-synchronization is detected (Case iii) above), synchronization algorithm is executed. For example, refer to Fig 5. The STA sends a frame(①) and 'ACK-success' is lost(F③) after the AP increases its pointer(②). The STA sends the frame again(④). The AP sends 'ACK-success'(S⑥) after comparing the authentication bit

values because they are same (STA's bit-1, AP's bit-2) despite of non-synchronization. The non-synchronization state continues until the authentication bit values differ. When the STA's bit-3 is sent to the AP(⑩), the AP becomes to notice non-synchronization and then runs the synchronization algorithm.

4 Performance Evaluation

The probability that an illegal user correctly guesses an n-bit authentication stream is 2^{-n}. Note that each bit of authentication stream is either 1 or 0, with the probability of average is 1/2.

Assume that a priori probability of STA to be an attacker is 50% such that P(illegal user) = 50% and P(legal user) = 50%. In case of no contiguous ACK loss, the probability of the STA being a legitimate one is found using Bayes' formula and binomial distribution. The Wang's scheme [1] describes an approach similar to this method.

Assume that the length of an authentication stream is N and n is the number of times synchronization was attempted using the stream due to lost packet. The probability that an ACK frame lost is p. P(STA = legal user | N,n) is the probability of legal user. Using Bayes' formula,

$$\begin{aligned} &\text{P(STA = legal user} \mid N,n) = 1 - \text{P(STA = illegal user} \mid N,n) \\ &= \frac{P(N,n \mid STA = legal\ \ user)}{P(N,n \mid STA = legal\ \ user) + P(N,n \mid STA = illegal\ \ user)} \end{aligned} \tag{1}$$

$$\text{P}(N,n \mid \text{STA = illegal user}) = \binom{N}{n} * 2^{-N} \tag{2}$$

Illegal user does not know the next bit, and thus randomly chooses zero or one.

$$\text{P}(N,n \mid \text{STA = legal user}) = \binom{N}{n} * p^{n}(1-p)^{N-n} \tag{3}$$

Combining (2),(3), it is easy to get the following.

$$\text{P(STA = legal user} \mid N,n) = \frac{p^{n}(1-p)^{N-n}}{2^{-N} + p^{n}(1-p)^{N-n}} \tag{4}$$

Fig. 6 shows the probability of an STA being a legal user. The analysis is for p = 0.1, 0.3 and 0.5 and N=10. For example, when the AP finds n = 4, the probability of legal user is about 0.493 for the frame lost rate of 30%.

We next evaluate the performance of the proposed algorithm. The simulation has been carefully designed in order to illustrate the behavior and performance of the protocol, and it was implemented using C language. The results of 10 runs are averaged with an authentication stream of 20,000 bits.

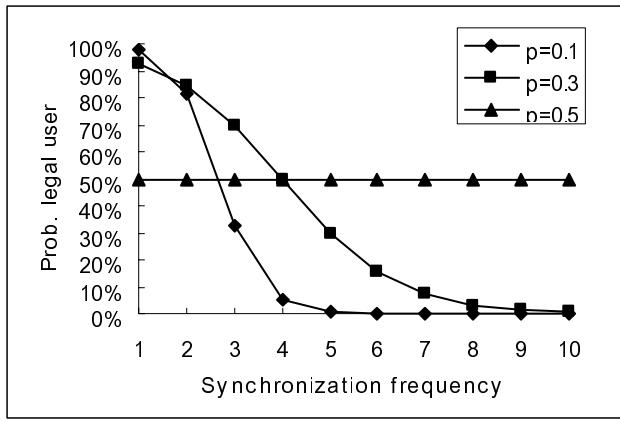


Fig. 6. The probability of legal user as n varies ($N = 10$).

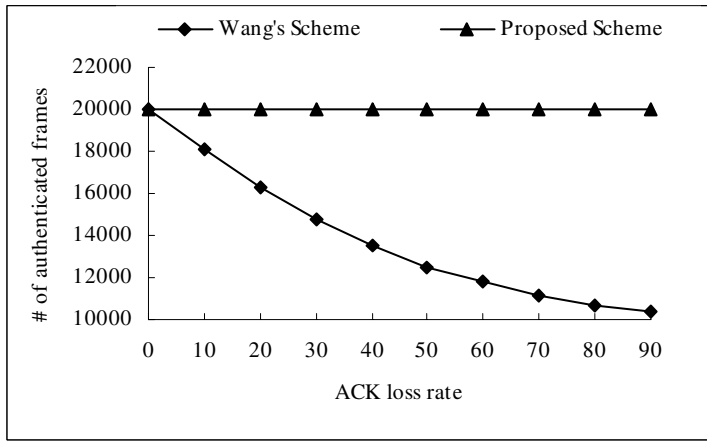


Fig. 7. Utilization of authentication stream with changing loss rate of ACK message .

Fig. 7 compares the number of authenticated frames of our scheme and Wang's scheme [1] as the loss rate of ACK message changes with the given 20,000 bit authentication stream. Compared with Wang's scheme, utilization of the stream with the proposed scheme is always 100% while it reduces significantly as the loss rate of ACK message grows with Wang's Scheme. For example, when the loss rate is 20%, the proposed scheme allows 20,000 authentications while Wang's scheme does that only about 16,000 authentications. Note that once the stream is used up, it needs to be generated again which is time consuming. Therefore, the proposed synchronization scheme allows much faster authentication. The new protocol can also be added without any change in the existing structure of IEEE 802.11.

5 Conclusion

In this paper a new efficient authentication protocol for access control in IEEE 802.11 networks has been presented. The proposed protocol attaches one authentication bit per frame obtained from a stream known only to the two communicating stations. It also employs an effective synchronization algorithm. Computer simulation reveals that proposed scheme significantly improves the throughput compared to the earlier protocol employing the same single bit authentication approach for practical operational environment. The proposed protocol also greatly reduces power consumption in the mobile station.

This work provides the basis for a new authentication protocol in wireless communication. The protocol could be very useful to provide secure communication in wireless environment. As a future work, we will analyze and compare response time and throughput of the proposed protocol with other protocols for various operational scenarios. Different authentication approaches will also be developed which will further enhance the performance.

Reference

[1] Hao-li Wang, Aravind Velayutham, and Yong Guan, "A Lightweight Authentication Protocol for Access Control in IEEE 802.11", submitted to IEEE GlobeCom, Mar. 2003.

[2] Henric Johnson, Arne Nilsson, Judy Fu, S.Felix Wu, Albert Chen and He Huang, "SOLA: A One-bit Identity Authentication Protocol for Access Control in IEEE802.11", In Proceedings of IEEE GLOBECOM, September 2002

[3] J. Walker, "Unsafe at any key size : an analysis of the WEP encapsulation", Tech. Rep. 03628E, IEEE 802.11 committee, March 2000.

[4] NORTEL networks com White paper : "Secure Architectures for Wireless LANs in the Enterprise" available from www.nortelnetworks.com

[5] "LAN MAN Standards of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specification IEEE Standard 802.11, 1997 Edition," 1997

[6] Institute of Electrical and Electronics Engineers(IEEE).Standard for port based Network Access Control. IEEE Draft p802.1X/D11. March 2001.

[7] Intel com "Intel Building Blocks for Wireless LAN Security" http://www.intel. com/ network/connectivity/resources/doc_library/white_papers/WLAN_Security WP.pdf#1-3, February 2003.

[8] N. Borisov, I. Goldberg, and D. Wangner, "Interception Mobile Communications: The Insecurity of 802.11"

[9] Yanyan Yang; Zhi Fu; Wu, S.F., "Bands: an inter-domain internet security policy management system for IPSEC/VPN", Integrated Network Management, IFIP/IEEE Eighth International Symposium on,2003.

[10] Bhagavathula R., Thanthry N., Pendse R., "Mobile IP and virtual private networks" Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th Sept. 2002

[11] CREWAVE CO.,Ltd, available from http://www.crewave.com/Korean/menu/support/tech/tech_1.htm

[12] Jingdi Zeng, Ansari, N., "Toward IP virtual private network quality of service: a service provider perspective", Communications Magazine, IEEE , Volume: 41 Issue: 4 , April 2003

[13] Intel, "VPN and WEP Wireless 802.11b security in a corporate environment", Intel white paper, March 2003

A Study on Secure and Efficient Sensor Network Management Scheme Using PTD

Dae-Hee Seo and Im-Yeong Lee

Division of Information Technology Engineering, SoonChunHyang University, #646, Eupnae-ri, Shinchang-myun, Asan-si, Coogchungnam-Do, 336-745, Republic of KOREA
Phone +82-41-542-8819, Fax +82-41-530-1548
{patima,imylee}@sch.ac.kr, http://sec-cse.sch.ac.kr

Abstract. Recently, many researchers have been focusing on ubiquitous computing, which is a new type of network environment. As one of the most important elements in ubiquitous computing, the sensor network consists of sensors based on low-powered ad-hoc networks and sensor nodes and acts as the medium between the real environment and ubiquitous computing. In this article, a secure and efficient sensor network structure is proposed after briefly examining the security of the sensor network and PTD analyzing its vulnerabilities.

Keywords: Ubiquitous Computing, Secure Sensor Network, Personal Trust Device

1 Introduction

A ubiquitous network environment optimizes the user's network environment to connect conveniently to the network by capturing the user's circumstances or environment intelligently. Likewise, it arranges a network to utilize contents freely and securely. The technologies composing a ubiquitous network include flexible broadband, teleportation, agent, contents, appliance, platform, and sensor network. Among these technologies, sensor network is an essential component collecting and managing information autonomously by communicating with peripheral equipment around users. Currently popular terminals such as PCs, PDAs, and hand phones have limited capabilities in terms of receiving information. In the future, if the minimized chips are distributed among pieces of equipment, the terminal can receive and save all kinds of information for free distribution depending on the demand of the users. By utilizing the minimized chips, the network can be managed by controlling the detailed information with the wireless electronic tag of the minimized chip in the bar-code management system[1]. As stated earlier, a sensor network collects and manages information autonomously. Chapter 2 presents an overview of the sensor network environment. In Chapter 3, we suggest the security requirement for the sensor network environment. In Chapter 4, a secure and efficient sensor network manage ment structure is suggested to satisfy the security requirements stated in Chapter 3. In Chapter 5, the suggestion is analyzed. The conclusions are presented in Chapter 6.

[1] Program for cultivating graduate in regional strategic industry

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3045, pp. 1020–1028, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Overview of Sensor Network and PTDs

Chapter 2 provides an overview of the technological introduction of sensor network and PTDs.

2.1 Overview of Sensor Network

Sensor network composes itself using sensor nodes distributed randomly in the space to measure and to transfer analog data, such as light in the physical space, sound, temperature, and motion of the object, to the centered node. The sensor nodes are generally the micro controllers using several MHz clocks, EEPROM with the capacity of dozen KBs, flash memory with several KBs, sensor elements (temperature, sound, light, degree of acceleration of object, magnetic field), output devices (LED, speaker), and communication modules(radio wave frequency).[1]

The sensor network is also known as the input network that converts the analog data measured in the physical space into the digital signal to transfer to the base node connected to the electronic space such as the Internet. The sensors in the network transfer the data with the type of agent. Therefore, the technologies required for the sensors should be able to formulate groups as automatically distributed objects and to provide users with services. Applied to the e-commerce environment, each sensor should not be stable; rather, they should have the ability to change depending on the circumstances. The sensors are also required to acquire information from other sensors to determine whether or not communication was set up. In addition, sensors can organize groups if the amount of communication exceeds such limits. [3]

Therefore, the sensors used in this environment will materialize with the availability of chips through advanced hardware technology and minimization technology. Nonetheless, the sensor network formulated around the users is closely related to the user's privacy. Therefore, security factors should be considered in this research.

2.2 Overview of PTDs

The development of the Internet accelerates with the mobile communication environment. With this trend, new technologies using mobile devices such as RUIM, GPS, MPEG4, and Bluetoooth are introduced and standardized.

Mobile devices process various data for users. They are also important hardware devices used in processing privacy data. Depending on the user's level and purposes, the mobile devices are expected to be subdivided and diversified from the simple voice communication to PDAs with high-speed computing capabilities. Therefore, with the realization of the unified environment satisfying the platform of the diverse devices and the open type design considering OEM, the service to protect the user's privacy should also be considered [2][4].

In general, PTDs mean cellular phones including the security modules. PTDs can be used not only to keep important data such as personal keys but also to authenticate personal information. Therefore, it can be used on the platform of application programs such as banking, payment, and bonus programs, among others. PTDs will

accomplish authentication and confirmation for the transferred data based on the user's ID. Thus, security elements are always required. To protect the user's privacy, the security service should be provided by hardware and software [5].

3 Necessity of the Security and Security Requirement of the Sensor Network

In this chapter, the necessity of the security and security requirement of the sensor network is discussed.

3.1 Necessity of Sensor Network Security

The sensor network is an essential factor in protecting not only the user's privacy but also business and society in general.

As far as the environment is concerned, everyone can access the user's information on the network. One of the weaknesses of the highly developed network environments is that the information can be hacked by a third party. This, in essence, is a cyber crime. Likewise, a small bug on the system can cause serious errors. There are also other negative effects aside from hacking, including being prone to viruses, invasion of privacy, and violation of copyright laws [2][6].

In the ubiquitous environment, if the private information is corrected or available illegally regardless of personal intension, the new network environment cannot be considered stable. This results in several social problems, which will be discussed in the later Chapter.

3.2 Analysis of Security Requirement Factors on the Sensor Network

If the sensor network is user-oriented, it is possible to communicate using private information. Because the information system has characteristics similar to those of sensors, it is important that it is able to collect information autonomously and to manage components on the PTDs. Still, lightweight, miniature sensors (low power level, lower calculation capacity, limited memory amount, etc) have limited capacity to process security operation.

Therefore, the following requirements should be satisfied to ensure the secure and efficient management of the sensor network using PTDs.

- Mutual authentication: The security of the sensor networks organized for users can be maintained by authenticating mobile terminals with the server or gateways.
- Confidentiality and integrity: To transfer the privacy information of users, the security of the transferred data should be maintained by ensuring the confidentiality and integrity of the transferred data.
    * Necessity of device security: If the device aims to protect confidential wireless traffic, the privacy of the information on the sensor network is very important. For example, if a PTDis lost, the PTD itself can be protected by a password. Nonetheless, the information stored in the PTD is merely saved

rather than coded. Therefore, an attacker with proper resources can browse and extract the stored information.

* Confidentiality of transmitted data: If the sensor network does not guarantee the confidentiality of message in transmitting and receiving data, personal privacy will be vulnerable.

– State acquisition technology: This determines the state of communication when the state of sensors changes. For this purpose, such technology should be provided to acquire information based on the secure interface with other sensors.
– Automatic service generation approach: If the frequency of usage is above a certain level, converting into a secure type group should be possible. If the frequency is below a certain level, an automatic group dismantling process should be provided.
– Efficiency: by minimizing the amount of calculation, we maximize the efficiency of the overall sensor network.

4 Proposed Sensor Network Structure for Secure and Efficient Management

This paper suggests a secure and efficient management structure of sensor networks based on home network. The proposed system is based on the following assumption.

– It supports several Mbps access link, and it is possible to connect to the network seamlessly inside or outside the buildings.
– All mobile devices are PTDs with User Agent. (UA is one of the service elements.)

4.1 Scenario of the Proposed System and Its Components

The proposed system offers a secure and efficient sensor network management structure that provides continuous services by generating and acquiring services through the collection of information for many users with wireless capability PTDs on the home network with no special wireless environment. Under this scenario, each object operates in the following manner:

① Users: Users have UA embedded PTDs. It is the object that generates services and acquires status after undergoing authentication in the server.
② Authentication server: It connects to the gateway to provide the services generated by the sensors. (The authentication server stores a list of sensors for initial distribution.)

4.2 System Parameters

n: Open coefficient ($n = pq$, p: Prime number, q: $q \mid p-1$)

m_i: This is the pre-defined generic number of service provided by the server.

$E()$, $D()$: Encryption, Decryption function

r, α, β: Random numbers

4.3 Protocol for the Proposed System

On the home network where there is no special wireless environment, if majority of mobile devices are PTDs, the proposed system has the following flow to compose the sensor network and to manage the structure securely and efficiently.

[Step1] Service registration process of SE
The authentication server performs the process to register the service of sensors using SE.

① The authentication server broadcasts c_i, which is calculated using randomly generated $\alpha \in_U Z_n$ to sensors.

$$c_i = m_i^{\alpha} \bmod n \ (i \text{ is service number, } i = 1, \ldots, \text{N})$$

② The sensor selects generic numbers of the service currently provided by UA from c_i transmitted from the server. If three services are selected, the sensor defines these numbers as c_{i_1}, c_{i_2} and c_{i_3}. After defining the type of service, the sensor selects and transfers $\beta \in_U Z_n$ ($\beta_1, \beta_2, \beta_3$)to the server after calculating d_{i_j}.

$$d_{i_j} = c_{i_j}^{\beta_*} \bmod n \ (= m_i^{\alpha\beta} \bmod n \ )(\ j = 1,2,3 \ )$$

③ After calculating $s = \alpha^{-1} \bmod n$, the authentication server transmits e_{i_j} to the sensor.

$$e_i = d_{i_j}^{s} \bmod n \ (= m_i^{\beta_*} \bmod n \ )(\ j = 1,2,3 \ )$$

④ After calculating $t = \beta_*^{-1} \bmod n$, the sensor confirms its propriety by verifying f_{i_j}.

$$f_{i_j} = e_{i_j}^{t} \bmod n \ (= m_i \ )(\ j = 1,2,3 \ )$$

[Step 2] Initial registration process of sensors for temporary group set up
The next step involves increasing the efficiency of service and communication by temporary grouping when more than a certain number of sensors require identical services.

① The sensor transmits Z_2 and T_s after calculating the following factors, in case the sensor wants to receive one of the services among c_{i_1}, c_{i_2} and c_{i_3} set up in [Step 1](to receive c_{i_2} service).

$$C_2 = c_{i_2} \oplus m_i \oplus ID_s, \ d_2 = c_{i_1} * c_{i_2} * r_{s_1}$$

$$Z_2 = C_2^{d_2} \bmod n$$

② The authentication server temporarily saves the value of Z_2 transmitted from the server. After calculating the temporary secret information value of d_n, Z_a and as

in the following equation, the server transmits Z_A, Z_a and T_a to the server after generating Z_A by selecting random number r_A from the authentication server:

$$d_n = c_{i_1} * c_{i_2} * c_{i_3}$$

$$Z_a = Z_2^{d_n^{-1}} \bmod n$$

$$Z_A = Z_a^{r_A} \bmod n$$

③ If Z_a, transmitted from the authentication server, is equal to V_2 ($Z_a \cong V_2$), the sensor transmits Z_2' and m_{i_2} to the authentication server, provided the calculation is correct.

$$V_2 = C_2^{c_{i2}^{-1} * r_{s1}} \bmod n$$

$$Z_2^{'} = Z_A^{r_{s1}} \bmod n$$

⑤ The authentication server saves y_2, the temporary secret information for the service required by sensors, using the value of m_{i_2} transmitted from the server.

$$y_2 = Z_2^{'\, r_A^{-1}} \bmod n$$

[Step 3] Set up process of temporary group

The authentication server confirms the secret information of y, which is registered in the sensor as stated in [Step 2], and m_*, transmitted as explained in [Step 1]. If the number of sensors receiving the same service exceeds a certain number, and if the frequency of requesting the same service is high, or if the same kind of communication increases, the authentication server executes the processes to set up a temporary group of corresponding sensors.

① The authentication server defines $(y_1,....,y_n)$ as the privacy information of the sensors, which waits to receive the same services. The server saves temporarily D_* generated for each y_*. To set up a temporary group (assuming ID_A, ID_B, ID_C are selected as a temporary group), s_A and T_A are transmitted to the sensor ID_A.

$$D_A = H(c_{i_2} \parallel y_A)$$

$$s_A = g^{D_A} \bmod n$$

② Sensor ID_A generates session key $K_{G_{temp}}$ using s_A transmitted from the authentication server. The sensor encrypted $K_{G_{temp}}$ by the application service message M_C as required for the secure communication using the generated session key ($K_{G_{temp}}$). After this process V_s and T_s are transmitted to the authentication server.

$$K_{G_{temp}} = s_A^{y_A^{-1}} \bmod n$$

$$V_S = E_{K_{temp}}(M_C) \bmod n$$

③ To confirm V_s which was transmitted, the authentication server verifies the application service needed for crypto communication after generating $K_{G_{temp}}$ and decrypting V_s. After confirming the service, the authentication server defines $K_{G_{temp}}$ as session keys with its temporary group members and secureguards the list $K_{G_{temp}}$.

$$K_{G_{temp}} = s_A^{y_A^{-1}} \bmod n$$

– The above-mentioned processes in [Step 3] are performed equally for all group members when setting up temporary groups.

[Step 4] Service requirement step of temporary group SE
The next step is to request the service for the representative sensor of the group to the server using the 1-out-2 oblivious transfer when the sensors, which are completed for the registration of the service, form a group to receive the same kind of service.

– Performing 1-out-2 oblivious transfer aims to increase the power efficiency of sensors by minimizing the possibilities of the loss of transmitted information or failure to transmit.

① The sensors broadcast random number r_0 and r_1 to the server.

② The server temporarily stores r_0 and r_1 received from the sensor. After selecting s_1 and x ($s_1 =_U \{0,1\}, x \in_U Z_n$) randomly, Q and T_A are calculated as shown below and transmitted to the server.

$$Q = E_{K_{temp}}(x) + r_{s_1} \bmod n$$

③ After calculating $x_{s_1} = D_{K_{temp}}(Q - r_i \bmod n)$, the sensor transmits c_0 and c_1 to the server after calculating as shown in the following (M_0 and M_1 are the service request messages for secure communication.):

$$c_0 = M_0 + x_0 \bmod n$$

$$c_1 = M_1 + x_1 \bmod n$$

④ The server acquires $M_{s_1} = c_{s_1} - x_{s_1} \bmod n$.

[Step 5] Deletion of temporary group
The authentication server performs the deletion process of temporary groups if the number of sensors forming such group falls below a certain number.

① The authentication server extracts a list D from temporarily stored information. This list is generated based on $(y_1,....,y_n)$, which is the secret information of sensors stored temporarily, and other secret information to form a temporary group.

② The authentication server broadcasts $s_{DEL} = (ID_A, D_1,....,ID_{DEL} * D_k)$ from the extracted value to the entire sensor network.

③ The sensor receiving the broadcasted information confirms the status of the temporary group that includes itself and checks the deletion of the temporary group.

By performing the above-mentioned steps, the protocol for secure and efficient sensor network structure is performed at the home network.

5 Analysis of the Proposed System

This paper proposes a secure and efficient network management process on the sensor network to achieve a ubiquitous environment. The suggested protocol provides a method that considers the capacity of the sensor by ensuring security to satisfy several security factors on the user-centered sensor network.

The proposed system for a secure and efficient sensor network management structure can maintain the following security status based on the security requirement for sensor network:

– Mutual authentication : For the sensor network organized for users, secure should be sustained through the mutual authentication of mobile terminals with the authentication server or gateway. Therefore, this paper utilizes the authentication system using the Feige-Fiat-Shamir authentication protocol. Still, the special characteristic of the Feige-Fiat-Shamir authentication protocol may make the size of the key very large. Therefore, selecting the right size of the key is crucial.
– Confidentiality and integrity: The security of the transmitted data should be maintained by providing data confidentiality and integrity when transmitting the user's private information.
  * Necessity of device security: SEs included in the objects on the home network should store the secret value securely as an object generating automatic services. As UA, SE preserves the confidentiality of the information saved in the device by performing the secret calculation in the calculation process on the sensor network. The suggested system performs a secret request calculation method by not transferring the secret value r_{s_1} of devices through the verification process of
  $$V_2 = C_2^{(c_{i_2}^{-1} * r_{s_1})} \bmod n \; (= Z_2^{d_n^{-1}} \bmod n) = C_2^{(c_{i_1} * c_{i_2} * r_{s_1}) * (c_{i_1} * c_{i_2} * c_{i_3})^{-1}} \bmod n .$$
  * Confidentiality of transferred data: On the sensor network of the home network, the confidentiality of the transferred data is maintained using the open key crypto algorithm and secure Hash function. $H()$
  * Integrity of transferred data: For the integrity of the transferred data, an integrity service is provided using the secure Hash function $H()$.
– Automatic service generation plan: if the frequency of services for communication exceeds a certain level, converting into the secure type of group should be possible. If the frequency of service falls below a certain level, an automatic group dismantling process should be provided. For the proposed system, the automatic service generation involves calculating f_{i_j} with i of the generic service for the service registration provided to the server. By specifying

the propriety of the service application, an automatic service generation method is provided.

– Efficiency: There is a need to increase the efficiency of the overall sensor network by minimizing the amount of calculation and by maximizing the efficiency of sensors. Therefore, to increase the efficiency of communication, this paper performs the oblivious transfer. By transmitting the crypto communication message c_0 and c_1, which are transmitted to the server by the sensor, another message $M_{s_1} (= c_{s_1} - x_{s_1} \bmod n)$ can be received even though one message is lost. Therefore, the size of the broadcasting message was minimized to transmit again.

6 Conclusion

Recent rapid development of telematique enables to increase the demand for personal telematique.

Under this environment, the sensor network can provide users with many conveniences in the future. If the security issues are not addressed, however, data will be prone to hackers, and user privacy will be compromised.

The sensor network in this kind of environment requires not only existing security factors such as authentication, confidentially, and integrity but also a new breed of security requirement for service. This paper proposes a secure and efficient sensor network management structure by satisfying existing and new security requirements.

Therefore, the proposed system structure can be utilized for e-commerce and other similar environments. Defining additional security factors and suggesting the solution enable the accomplishment of more advanced security services on the sensor network.

References

1. http://www.sktelecom.com//tlab/pdf/tr/13_1/13_1_07.pdf
2. http://user.chollian.net/~zmnlks/paper/reliable.pdf
3. http://citeseer.nj.nec.com/perrig01spins.html
4. http://citeseer.nj.nec.com/chen00security.html
5. http://citeseer.nj.nec.com/503260.html
6. http://citeseer.nj.nec.com/tolvanen00device.html
7. http://citeseer.nj.nec.com/tilak02taxonomy.html
8. http://citeseer.nj.nec.com/correal01wireless.html
9. Dae-Hee Seo, Im-Yeong Lee, Dong-ik Oh and Doo-Soon Park, "Bluetooth piconet using non-anonymous group key", Euraisa-ICT 2002, 2002.
10. Alfred J. Menezes, Paul C.van Oorschot and Scott A. Vanstone "HANDBOOK of APPLIED CRYPTOGRAPHY", CRC.

Author Index

Abawajy, J.H. II-107
Abawajy, Jemal II-87
Abdullah, Azizol II-146
Abellanas, Manuel III-1, III-22
Acciani, Giuseppe II-979
Acosta-Elías, Jesús IV-177
Aggarwal, J.K. IV-311
Ahmad, Muhammad Bilal IV-877, IV-940, IV-948
Ahn, Byoungchul III-566, III-993
Ahn, Byungjun I-1125
Ahn, In-Mo IV-896
Ahn, Jaemin III-847
Ahn, JinHo III-376, IV-233
Ahn, Kiok I-1044
Ahn, ManKi I-517
Ahn, Seongjin I-142, I-1078
Ahn, Sung IV-489
Ahn, Yonghak I-1044
Ahn, Young Soo II-1079
Albertí, Margarita II-328, II-374
Albrecht, Andreas A. III-405
Alcaide, Almudena I-851
Alegre, David III-857
Aleixos, Nuria II-613
Alinchenko, M.G. III-217
Amaya, Jorge II-603
An, Beongku IV-974
An, Changho I-25
An, Ping IV-243
Anido, Luis II-922
Anikeenko, A.V. III-217
Annibali, Antonio III-722
Apu, Russel A. II-592
Asano, Tetsuo III-11
Atiqullah, Mir M. III-396
Attiya, Gamal II-97
Aung, Khin Mi Mi IV-574

Bachhiesl, Peter III-538
Bae, Hae-Young I-222, II-1079
Bae, Ihn-Han I-617
Bae, Sang-Hyun I-310, II-186, IV-359
Baik, Kwang-ho I-988
Baik, Ran III-425
Baik, Sung III-425, IV-206, IV-489
Bajuelos, António Leslie III-117, III-127
Bala, Jerzy IV-206, IV-489
Bang, Young-Cheol I-1125, II-913, IV-56
Bang, Young-Hwan I-491
Barel, Marc Van II-932
Barenco Abbas, Clàudia Jacy I-868
Barua, Sajib III-686
Becucci, M. II-374
Bekker, Henk III-32
Bellini, Francesco III-722
Beltran, J.V. II-631
Bencsura, Ákos II-290
Bertazzon, Stefania II-998
Bhatt, Mehul III-508
Bollman, Dorothy III-481, III-736
Boluda, Jose A. IV-887
Bonetto, Paola II-505
Bonitz, M. II-402
Borgosz, Jan III-715, IV-261
Borruso, Giuseppe II-1009, II-1089
Bose, Prosenjit III-22
Botana, F. II-761
Brass, Peter III-11
Brink, Axel III-32
Broeckhove, Jan IV-514
Brunelli, Roberto II-693
Bruno, D. II-383
Bruschi, Viola II-779
Bu, Jiajun III-886, IV-406
Bücker, H. Martin II-882
Buliung, Ronald N. II-1016
Buono, Nicoletta Del II-961, II-988
Buyya, Rajkumar IV-147
Byun, Kijong II-809

Cacciatore, M. II-366
Caeiro, Manuel II-922
Camp, Ellen Van II-932
Campa, S. II-206
Campos-Canton, Isaac IV-177
Capitelli, Francesco II-338
Capitelli, M. II-383

Carbonell, Mildrey I-903
Carretero, Jesús IV-496
Carvalho, Sílvia II-168
Casas, Giuseppe Las II-1036
Cendrero, Antonio II-779
Čermák, Martin III-325
Cha, Eui-Young II-486, IV-421
Cha, JeongHee I-17, I-41
Cha, Joo-Heon II-573
Chae, Jongwoo III-965, IV-983
Chae, Kijoon I-673
Chae, Oksam I-1044
Chambers, Desmond II-136
Chang, Beom H. I-191, I-693, IV-681
Chang, Byeong-Mo I-106
Chang, Hoon I-73
Chang, Min Hyuk IV-877
Chang, Yongseok IV-251
Chelli, R. II-374
Chen, Chun III-886, IV-406
Chen, Deren II-158
Chen, Tzu-Yi IV-20
Chen, Yen Hung III-355
Chen, Zhenming III-277
Cheng, Min III-729
Cheung, Chong-Soo I-310
Cheung, Wai-Leung II-246
Chi, Changkyun IV-647
Cho, Cheol-Hyung II-554, III-53
Cho, Chung-Ki III-847, III-926
Cho, Dong-Sub III-558
Cho, Haengrae III-548, III-696
Cho, Hanjin I-1007
Cho, Jae-Hyun II-486, IV-421
Cho, Jeong-Hyun IV-359
Cho, Jung-Hyun IV-251
Cho, Kyungsan I-167
Cho, Mi Gyung I-33
Cho, Seokhyang I-645
Cho, SungEon I-402
Cho, TaeHo I-567
Cho, We-Duke I-207, I-394
Cho, Yongsun I-426
Cho, Yookun I-547, I-978, IV-799
Cho, Youngjoo IV-647
Cho, Youngsong II-554, III-62
Choi, Chang-Gyu IV-251
Choi, Chang-Won I-302
Choi, Changyeol I-207
Choi, Dong-Hwan III-288
Choi, Doo Ho I-1151
Choi, Eun-Jung I-683
Choi, Eunhee II-913
Choi, Hoo-Kyun IV-11
Choi, Hoon II-196
Choi, HyungIl I-17, I-41
Choi, Joonsoo III-837
Choi, Kee-Hyun I-434
Choi, SangHo IV-29
Choi, Sung Jin IV-637
Choi, Tae-Sun IV-271, IV-291, IV-338, IV-348, IV-877
Choi, Uk-Chul IV-271
Choi, Won-Hyuck IV-321, IV-451
Choi, Yong-Soo I-386
Choi, Yoon-Hee IV-271, IV-338, IV-348
Choi, YoungSik I-49, II-942
Choi, Yumi I-663
Choirat, Christine III-298
Chong, Kiwon I-426
Choo, Hyunseung I-360, I-663, I-765, III-315, IV-56, IV-431
Choo, Kyonam III-585
Chover, M. II-622, II-703
Choy, Yoon-Chul IV-743, IV-772
Chu, Jie II-126
Chun, Jong Hun IV-940
Chun, Junchul I-25
Chun, Myung Geun I-635, IV-828, IV-924
Chung, Chin Hyun I-1, I-655, IV-964
Chung, Ilyong II-178, IV-647
Chung, Jin Wook I-142, I-1078
Chung, Min Young I-1159, IV-46
Chung, Mokdong I-537, III-965, IV-983
Chung, Tai-Myung I-183, I-191, I-238, I-693, IV-681
Cintra, Marcelo III-188
Clifford, Gari I-352
Collura, F. II-536
Contero, Manuel II-613
Costa Sousa, Mario III-247
Crane, Martin III-473
Crocchianti, Stefano II-422
Crothers, D.S.F. II-321
Cruz R., Laura III-415, IV-77

Cruz-Chavez, Marco Antonio IV-553
Cutini, V. II-1107
Cyganek, Bogusław III-715, IV-261

D'Amore, L. II-515
Dağ, Hasan III-795
Daly, Olena IV-543
Daněk, J. II-456
Danelutto, M. II-206
Das, Sandip III-42
Datta, Amitava IV-479
Datta, Debasish IV-994
Delaitre, T. II-30
Demidenko, Eugene IV-933
Denk, F. II-456
Díaz, José Andrés III-158
Díaz-Báñez, Jose Miguel III-99, III-207
Díaz-Verdejo, Jesús E. I-841
Diele, Fasma II-932, II-971
Discepoli, Monia III-745, IV-379
Djemame, Karim II-66
Dong, Zhi II-126
Dózsa, Gábor II-10
Duato, J. II-661
Durán, Alfonso I-949, III-857

Effantin, Brice III-648
Eick, Christoph F. IV-185
Engel, Shane II-1069
Eom, Sung-Kyun IV-754
Ercan, M. Fikret II-246
Erciyes, Kayhan III-518, III-528
Esposito, Fabrizio II-300
Estévez-Tapiador, Juan M. I-841
Estrada, Hugo IV-506, IV-783
Eun, Hye-Jue I-122

Fan, Kaiqin II-126
Farias, Cléver R.G. de II-168
Faudot, Dominique III-267
Feng, Yu III-498
Fernández, Marcos II-661, II-671
Fernández-Medina, Eduardo I-968
Ferrer-Gomila, Josep Lluís I-831, I-924, IV-223
Filinov, V. II-402
Fiori, Simone II-961
Flahive, Andrew III-508
Formiconi, Andreas Robert II-495
Fornarelli, Girolamo II-979
Fortov, V. II-402
Foster, Kevin III-247
Fragoso Diaz, Olivia G. IV-534, IV-808
Fraire H., Héctor III-415, IV-77
Frausto-Solís, Juan III-415, III-755, IV-77, IV-553
Fung, Yu-Fai II-246

Galpert, Deborah I-903
Gálvez, Akemi II-641, II-651, II-771, II-779
Gameiro Henriques, Pedro II-817
García, Alfredo III-22
Garcia, Ernesto II-328
García, Félix IV-496
García, Inmaculada III-877
García, José Daniel IV-496
García-Teodoro, Pedro I-841
Gardner, Henry III-776
Gavrilova, Marina L. II-592, III-217
Gerace, Ivan III-745, IV-379
Gerardo, Bobby D. I-97
Gervasi, Osvaldo II-827, II-854
Giansanti, Roberto III-575
Go, Hyoun-Joo IV-924
Gola, Mariusz III-611
Gómez, Francisco III-207
González Serna, Juan G. IV-137
Gourlay, Iain II-66
Goyeneche, A. II-30
Gregori, Stefano II-437
Grein, Martin II-843
Guan, Jian III-706
Guarracino, Mario R. II-505, II-515
Gulbag, Ali IV-389
Guo, Wanwu IV-471, IV-956
Guo, Xinyu II-751
Gupta, Sudhir IV-791
Gutiérrez, Carlos I-968
Gutiérrez, Miguel III-857

Ha, Eun-Ju IV-818
Ha, JaeCheol I-150
Ha, Jong-Eun IV-896, IV-906, IV-915
Ha, Kyeoung Ju IV-196
Ha, Yan I-337
Hackman, Mikael I-821
Hahn, Kwang-Soo III-837

Hamam, Yskandar II-97
Hamdani, Ajmal H. II-350
Han, Dongsoo IV-97
Han, Jongsu III-955
Han, Qianqian II-272
Han, Seok-Woo I-122
Han, Seung Jo IV-948
Han, Sunyoung I-1115
Han, Tack-Don II-741
Han, Young J. I-191, I-693, IV-681
Haron, Fazilah IV-147
Healey, Jennifer I-352
Heo, Joon I-755
Herges, Thomas III-454
Hernández, Julio César I-812, I-851, I-960
Hiyoshi, Hisamoto III-71
HlaváII-ček, I. II-456
Hlavaty, Tomas III-81
Hoffmann, Kenneth R. III-277
Hong, Choong Seon I-755, I-792, I-915, I-1134
Hong, Chun Pyo III-656, IV-106
Hong, Dong Kwon I-134
Hong, Hyun-Ki II-799
Hong, Inki I-1125
Hong, Kwang-Seok I-89, IV-754
Hong, Man-Pyo IV-611
Hong, Manpyo III-867, IV-708
Hong, Maria I-57
Hong, Seong-sik I-1060
Hong, Suk-Ki II-902, II-913
Hong, Youn-Sik III-1002
Hosseini, Mohammad Mahdi III-676
Hruschka, Eduardo R. II-168
Hu, Hualiang II-158
Hu, Weixi II-751
Huang, Changqin II-158
Huettmann, Falk II-1117
Huguet-Rotger, Llorenç I-831, IV-223
Huh, Eui-Nam I-370, I-738, I-746
Hur, Hye-Sun III-1002
Hurtado, Ferran III-22
Hwang, Byong-Won III-386, IV-281
Hwang, Chan-Sik III-288
Hwang, Chong-Sun I-286, III-945, IV-233, IV-584
Hwang, EenJun IV-838, IV-859
Hwang, Ha Jin I-577
Hwang, Jun I-1, I-655, I-746
Hwang, Seong Oun II-46
Hwang, Sun-Myung I-481
Hwang, Sungsoon II-1026
Hwang, Yong Ho I-442
Hwang, Yong-Ho II-799
Hwang, YoungHa IV-460

Ibrahim, Hamidah II-146
Iglesias, A. II-641, II-651, II-771
Im, Chaetae I-246
Im, Jae-Yuel IV-655
In, Chi Hyung I-792
Inguglia, Fabrizio II-505
Izquierdo, Antonio I-812

Jabbari, Arash II-432
Jacobs, Gwen III-257
Jang, HyoJong I-41
Jang, Jong-Soo I-988, IV-594
Jang, Jongsu I-776
Jang, Kyung-Soo I-434
Jang, Min-Soo III-489
Jang, Sang-Dong II-216
Jang, Seok-Woo I-9
Jang, Tae-Won I-386
Je, Sung-Kwan IV-421, II-486
Jedlovszky, P. III-217
Jeon, Hoseong I-765
Jeon, Jaeeun III-566
Jeong, Chang Yun I-337
Jeong, Chang-Sung I-319, II-789
Jeong, Eunjoo I-418
Jeong, Hae-Duck J. III-827
Jeong, Ok-Ran III-558
Jeong, Sam Jin IV-213
Jiang, Minghui III-90
Jin, Guiyue III-993
Jin, Hai II-116, II-126
Jin, Min IV-763, IV-849
Jin, Zhou II-272
Jo, Hea Suk I-711, III-1010
Jo, Jang-Wu I-106
Jo, Sun-Moon IV-524
Jonsson, Erland I-821
Jonsson, Håkan III-168
Joo, Pan-Yuh I-394
Jorge, Joaquim II-613

Jun, Woochun II-902, II-913
Jung, Changryul I-294
Jung, Il-Hong I-451
Jung, Kyung-Yong II-863
Jung, Yoon-Jung I-491

Kacsuk, Péter II-10, II-37, II-226
Kanaroglou, Pavlos S. II-1016
Kang, Chang Wook II-554
Kang, Dong-Joong IV-896, IV-906, IV-915
Kang, Euisun I-57
Kang, HeeGok I-402
Kang, Ho-Kyung III-602
Kang, Ho-Seok I-1105
Kang, Hyunchul I-345
Kang, Kyung-Pyo IV-348
Kang, KyungWoo I-65
Kang, Min-Goo I-302, I-386, I-394
Kang, SeokHoon I-270, III-585
Kang, Seung-Shik IV-735
Kang, Sunbu III-926
Kang, Sung Kwan IV-940
Kang, Sungkwon III-847, IV-11
Kang, Tae-Ha IV-281
Kang, Won-Seok IV-167
Kasahara, Yoshiaki I-915
Kasprzak, Andrzej III-611
Kaußner, Armin II-843
Kelz, Markus III-538
Kheddouci, Hamamache III-267
Kim, Backhyun I-345
Kim, Bonghan I-1007
Kim, Byoung-Koo I-998, IV-594
Kim, Byunggi I-418
Kim, Byungkyu III-489
Kim, Chang Hoon III-656, IV-106
Kim, Chang-Soo I-410
Kim, ChangKyun I-150
Kim, Changnam I-738
Kim, ChaYoung IV-233
Kim, Cholmin III-867
Kim, D.S. I-183
Kim, Dae Sun I-1134
Kim, Dae-Chul IV-271
Kim, Daeho I-1078
Kim, Deok-Soo II-554, II-564, II-583, III-53, III-62
Kim, Dohyeon IV-974
Kim, Dong S. I-693, IV-681
Kim, Dong-Hoi I-81
Kim, Dong-Kyoo III-896, III-906, IV-611
Kim, Dongho I-57
Kim, Donguk III-62
Kim, Duckki I-378
Kim, Gwang-Hyun I-1035
Kim, Gyeyoung I-9, I-17, I-41
Kim, Haeng-Kon I-461
Kim, Haeng-kon IV-717
Kim, Hak-Ju I-238
Kim, Hak-Keun IV-772
Kim, Hangkon I-587
Kim, Hanil II-892
Kim, Hie-Cheol II-20
Kim, Hiecheol III-656
Kim, Ho J. IV-791
Kim, Hyeong-Ju I-998
Kim, Hyun Gon I-1151
Kim, Hyun-Sung IV-617
Kim, Hyuncheol I-1078
Kim, Hyung-Jong I-567, I-683
Kim, Ik-Kyun I-998, IV-594
Kim, Iksoo I-270, I-345
Kim, Injung I-491
Kim, Jae-Kyung IV-743
Kim, Jaehyoun I-360
Kim, Jay-Jung II-573
Kim, Jeeyeon I-895
Kim, Jeom Goo I-1026
Kim, Jin I-81
Kim, Jin Geol IV-29
Kim, Jin Ok I-1, I-655, IV-964
Kim, Jin Soo IV-964
Kim, Jong G. II-1
Kim, Jong-bu IV-725
Kim, Jong-Woo I-410
Kim, Joo-Young IV-338
Kim, JoonMo I-567
Kim, Jung-Sun I-175, III-985, IV-321, IV-451
Kim, Jung-Woo II-741
Kim, Kee-Won IV-603, IV-672
Kim, Keecheon I-1115
Kim, Ki-Hyung IV-167
Kim, Ki-Tae IV-524
Kim, Ki-Young I-988, IV-594

Kim, KiIl IV-460
Kim, KiJoo I-49
Kim, Kweon Yang I-134
Kim, Kyungsoo II-467
Kim, Mansoo I-537
Kim, Mi-Ae I-159, I-722
Kim, Mi-Jeong I-394
Kim, Mihui I-673
Kim, Min-Su I-1159
Kim, Minsoo I-175, I-230
Kim, Misun I-199, I-262
Kim, Miyoung I-199, I-262
Kim, MoonJoon I-73
Kim, Moonseong IV-56
Kim, Myuhng-Joo I-683
Kim, Nam-Chang I-1105
Kim, Nam-Yeun IV-87
Kim, Pan Koo IV-940
Kim, Pankoo II-892
Kim, Pyung Soo III-975, IV-301
Kim, Sang Ho I-608, I-1069
Kim, SangHa IV-460
Kim, Sangkyun I-597
Kim, Seokyu I-150
Kim, Seong-Cheol III-837
Kim, Seonho I-328
Kim, Seungjoo I-645, I-895
Kim, Shin-Dug II-20
Kim, Soon Seok I-215
Kim, Soon-Dong IV-611
Kim, Soung Won I-577
Kim, Su-Hyun I-1035
Kim, Sung Jo I-278
Kim, Sung Ki I-246
Kim, Sung Kwon I-215
Kim, Sung-Ho IV-251
Kim, Sung-Hyun I-150
Kim, Sung-Min III-602
Kim, Sung-Ryul III-367
Kim, Sung-Suk IV-924
Kim, Sunghae I-1078
Kim, Sungsoo I-207
Kim, SungSuk I-286
Kim, Tae-Kyung I-238
Kim, Taekkeun III-926
Kim, Tai-Hoon I-451, I-461, I-1052, IV-717
Kim, Won I-17
Kim, Wonil III-896, III-906
Kim, Woo-Hun IV-617
Kim, Wu Woan II-216, II-262
Kim, Yong-Guk III-489
Kim, Yong-Sung I-122, I-337
Kim, Yoon Hyuk II-467
Kim, Young Kuen III-975
Kim, Young-Chon IV-994
Kim, Young-Sin I-738, I-746
Kim, YounSoo II-196
Kiss, T. II-30
Kizilova, Natalya II-476
Ko, Myeong-Cheol IV-772
Ko, Younghun I-360
Kóczy, László T. I-122
Koh, JinGwang I-294, I-310, I-402
Koh, Kwang-Won II-20
Kolingerová, Ivana II-544, II-682, III-198
Koo, Han-Suh II-789
Kouadri Mostéfaoui, Ghita I-537, III-965, IV-983
Kouh, Hoon-Joon IV-524
Ku, Kyo Min IV-196
Kulikov, Gennady Yu. III-345, III-667
Kwak, JaeMin I-402
Kwak, Jin I-895, III-955
Kwak, Keun Chang I-635, IV-828, IV-924
Kwon, Chang-Hee I-310
Kwon, Ki Jin I-1159
Kwon, Kyohyeok I-142
Kwon, Soonhak III-656, IV-106
Kwon, Taekyoung I-728
Kwon, Yong-Won I-319
Kwon, YongHoon III-847, III-926

Laccetti, G. II-515, II-525
Laganà, Antonio II-328, II-357, II-374, II-422, II-437, II-827, II-854
Lagzi, István II-226
Lang, Bruno II-882
Lara, Sheila L. Delfín IV-808
Lau, Matthew C.F. II-873
Lázaro, Miguel II-779
Lee, Bo-Hyeong IV-46
Lee, Bong Hwan I-352
Lee, Bum Ro IV-964
Lee, Byong Gul I-134

Lee, Byong-Lyol I-663
Lee, Byung Kwan I-33
Lee, Byung-Wook I-746
Lee, Byunghoon III-53
Lee, Dae Jong I-635, IV-828
Lee, Dea Hwan I-915
Lee, Deok-Gyu IV-66
Lee, Dong Chun I-1052, I-1097
Lee, Dongkeun I-1115
Lee, Dongryeol I-510
Lee, Eun-ser I-451
Lee, Gang-Soo I-491
Lee, Gunhee III-906
Lee, Gunhoon III-566
Lee, Hae-Joung IV-994
Lee, Hae-ki IV-725
Lee, Han-Ki I-159
Lee, Ho-Dong III-489
Lee, HongSub I-567
Lee, HoonJae I-517
Lee, Hunjoo II-809, II-837
Lee, Hwang-Jik II-20
Lee, Hyon-Gu I-89
Lee, Hyun Chang II-186
Lee, HyunChan II-554, II-564
Lee, Hyung-Woo I-302, I-386
Lee, HyungHyo I-701
Lee, Im-Yeong I-557, III-1020, IV-66
Lee, In Hwa I-278
Lee, In-Ho II-573
Lee, Jae Kwang I-254, I-1007
Lee, Jae-il I-728
Lee, Jaeheung I-547
Lee, Jaeho II-564
Lee, Jaewan I-97
Lee, Jong Sik III-621, III-630
Lee, Jong-Suk Ruth III-827
Lee, Joongjae I-17
Lee, Ju-Hyun IV-11
Lee, Jung-Hyun II-863
Lee, Jungsik I-97
Lee, KangShin I-567
Lee, Keon-Jik III-638
Lee, Key Seo IV-964
Lee, Ki Dong III-566, III-993
Lee, Kwan H. III-178
Lee, Kwang-Ok I-310
Lee, Kwnag-Jae IV-451
Lee, Kyong-Ho IV-743
Lee, Kyung Whan I-451
Lee, Malrey I-97
Lee, Myung Eui III-975, IV-301
Lee, Myung-Sub IV-441
Lee, Namhoon I-491
Lee, Okbin II-178
Lee, Ou-Seb I-394
Lee, Pil Joong I-442, I-471, I-802
Lee, Sang Hyo IV-964
Lee, Sang-Hak III-288
Lee, Sang-Ho IV-689
Lee, Sangkeon I-1017, I-1088
Lee, SangKeun I-286
Lee, Seok-Joo III-489
Lee, Seung IV-725
Lee, SeungYong I-701
Lee, Soo-Gi I-625
Lee, SooCheol IV-838, IV-859
Lee, Soung-uck III-867
Lee, Sung-Woon IV-617
Lee, Sungchang I-1125
Lee, Sungkeun I-294
Lee, Tae-Jin I-1159, IV-46
Lee, Tae-Seung III-386, IV-281
Lee, Taehoon II-178
Lee, Tong-Yee II-713, II-721
Lee, Won Goo I-254
Lee, Won-Ho III-638
Lee, Won-Hyung I-159, I-722
Lee, Won-Jong II-741
Lee, Woojin I-426
Lee, Woongjae I-1, I-655
Lee, YangKyoo IV-838, IV-859
Lee, Yeijin II-178
Lee, YoungSeok II-196
Lee, Yugyung I-410
Leem, Choon Seong I-597, I-608, I-1069
Lendvay, György II-290
Levashov, P. II-402
Lho, Tae-Jung IV-906
Li, Chunlin IV-117
Li, Gang II-252
Li, Layuan IV-117
Li, Mingchu II-693
Li, Shengli II-116
Li, Xiaotu II-252
Li, Xueyao IV-414

Li, Yufu II-116
Lim, Heeran IV-708
Lim, Hwa-Seop I-386
Lim, Hyung-Jin I-238
Lim, Joon S. IV-791
Lim, SeonGan I-517
Lim, Soon-Bum IV-772
Lim, Younghwan I-57
Lin, Hai II-236
Lin, Ping-Hsien II-713
Lin, Wenhao III-257
Lindskog, Stefan I-821
Lísal, Martin II-392
Liturri, Luciano II-979
Liu, Da-xin III-706
Liu, Yongle III-498
Llanos, Diego R. III-188
Lombardo, S. II-1046
Longo, S. II-383
Lopez, Javier I-903
López, Mario Alberto III-99
Lovas, Róbert II-10, II-226
Lu, Chaohui IV-243
Lu, Jianfeng III-308
Lu, Yilong III-729
Lu, Yinghua IV-956
Lu, Zhengding IV-117
Luna-Rivera, Jose Martin IV-177
Luo, Yingwei III-335

Ma, Zhiqiang IV-471
Machì, A. II-536
Maddalena, L. II-525
Maponi, Pierluigi III-575
Marangi, Carmela II-971
Mariani, Riccardo III-745
Marinelli, Maria III-575
Mark, Christian II-843
Marques, Fábio III-127
Marshall, Geoffrey III-528
Martínez, Alicia IV-506, IV-783
Martoyan, Gagik A. II-313
Mastronardi, Nicola II-932
Matsuhisa, Takashi III-915
Maur, Pavel III-198
Medvedev, N.N. III-217
Mejri, Mohamed I-938
Melnik, Roderick V.N. III-817
Ménegaux, David III-267
Merkulov, Arkadi I. III-667
Merlitz, Holger III-465
Messelodi, Stefano II-693
Miguez, Xochitl Landa IV-137
Milani, Alfredo III-433, IV-563
Min, Byoung Joon I-246
Min, Hongki III-585
Min, Jun Oh I-635, IV-828
Min, Young Soo IV-869
Minelli, P. II-383
Ming, Zeng IV-127
Mitrani, I. II-76
Moh, Sangman IV-97
Molina, Ana I. III-786
Mollá, Ramón III-877
Monterde, J. II-631
Moon, Aekyung III-696
Moon, Kiyoung I-776
Moon, SangJae I-150, I-517
Moon, Young-Jun I-1088
Mora, Graciela IV-77
Moradi, Shahram II-432
Moreno, Oscar III-481
Moreno-Jiménez, Carlos III-1
Morici, Chiara III-433
Morillo, P. II-661
Mukherjee, Biswanath IV-994
Mumey, Brendan III-90
Mun, Youngsong I-199, I-262, I-378, I-738, I-1144
Murgante, Beniamino II-1036
Murli, A. II-515
Murri, Roberto III-575
Muzaffar, Tanzeem IV-291

Na, Jung C. I-191, I-693, IV-681
Na, Won Shik I-1026
Na, Young-Joo II-863
Nam, Dong Su I-352
Nam, Junghyun I-645
Nandy, Subhas C. III-42
Navarro-Moldes, Leandro IV-177
Naya, Ferran II-613
Nedoma, Jiří II-445, II-456
Neelamkavil, Francis II-741, IV-743
Németh, Csaba II-10
Nguyen, Thai T. IV-791
Nicotra, F. II-536

Nielsen, Frank III-147
Niewiadomski, Radoslaw III-433
Nishida, Tetsushi III-227
Nock, Richard III-147
Noh, Bong-Nam I-175, I-230
Noh, BongNam I-701
Noh, JiSung II-942
Noh, SungKee IV-460
Noltemeier, Hartmut II-843

O'Loughlin, Finbarr II-136
O'Rourke, S.F.C. II-321
Oh, Am Sok I-33
Oh, ByeongKyun I-527, IV-698
Oh, Jai-Ho I-765
Oh, Kyu-Tae III-985
Oh, Soohyun III-955
Oh, Sun-Jin I-617
Oh, Wongeun I-294
Oh, Young-Hwan I-222
Ohn, Kyungoh III-548
Olanda, Ricardo II-671
Oliveira Albuquerque, Robson de I-868
Onieva, Jose A. I-903
Orduña, J.M. II-661
Orozco, Edusmildo III-481, III-736
Orser, Gary III-257
Ortega, Manuel III-786
Otero, César II-641, II-779, III-158
Othman, Abdulla II-66
Othman, Abu Talib II-146
Othman, Mohamed II-146
Ouyang, Jinsong I-345
Ozturk, Zafer Ziya IV-398

Pacifici, Leonardo II-357
Pakdel, Hamid-Reza III-237
Palladini, Sergio II-1057
Palmer, J. II-76
Palmieri, Francesco I-882
Palop, Belén III-188
Pan, Zhigeng II-236, II-731, II-751, III-308
Pardo, Fernando IV-887
Park, Chang Won IV-627
Park, Chang-Hyeon IV-441
Park, Dong-Hyun II-863
Park, Goorack I-25
Park, Gwi-Tae III-489
Park, Gyung-Leen I-114
Park, Hee-Un I-557
Park, Hong Jin I-215
Park, Hyoung-Woo I-319, II-1, III-827
Park, Hyunpung III-178
Park, IkSu I-527, IV-698
Park, JaeHeung I-73
Park, Jaehyung I-1159
Park, Jihun IV-311, IV-369
Park, Jong An IV-877, IV-940, IV-948
Park, Jong Sou IV-574
Park, Jongjin I-1144
Park, Joo-Chul I-9
Park, Joon Young II-554, II-564
Park, Jun-Hyung I-230
Park, Ki heon IV-29
Park, Kyeongmo I-500
Park, Kyung-Lang II-20
Park, M.-W. II-573
Park, Mingi I-97
Park, Namje I-776
Park, Sangjoon I-418
Park, Seong-Seok I-410
Park, Seung Jin IV-877, IV-948
Park, SeungBae I-527, IV-698
Park, Sihn-hye III-896
Park, Soohong III-975
Park, Soon-Young II-1079
Park, Sunghun IV-311, IV-369
Park, Taehyung I-1017, I-1088
Park, Taejoon II-837
Park, Woo-Chan II-741
Park, Yongsu I-547, I-978, IV-799
Pastor, Oscar IV-506, IV-783
Payeras-Capella, Magdalena I-831, IV-223
Pazos R., Rodolfo A. III-415, IV-77
Pedlow, R.T. II-321
Peña, José M. II-87
Pérez O., Joaquín III-415, IV-77
Pérez, José María IV-496
Pérez, María S. II-87
Pérez, Mariano II-671
Petri, M. II-1046, II-1107
Petrosino, A. II-525
Pfarrhofer, Roman III-538
Pflug, Hans-Joachim II-882
Piantanelli, Anna III-575

Piattini, Mario I-968
Pieretti, A. II-366
Piermarini, Valentina II-422
Pierro, Cinzia II-338
Pietraperzia, G. II-374
Pineda, Ulises IV-177
Ping, Tan Tien IV-147
Pişkin, Şenol III-795
Podesta, Karl III-473
Poggioni, Valentina IV-563
Politi, Tiziano II-961, II-988
Ponce, Eva I-949
Porschen, Stefan III-137
Puchala, Edward IV-39
Pugliese, Andrea II-55
Puig-Pey, J. II-651, II-771
Puigserver, Macià Mut I-924
Puttini, Ricardo S. I-868

Qi, Zhaohui II-252
Qin, Zhongping III-90

Ra, In-Ho I-310, IV-359
Radulovic, Nenad III-817
Ragni, Stefania II-971
Rahayu, Wenny III-443, III-508
Ramos, J.F. II-622, II-703
Ramos, Pedro III-22
Rebollo, C. II-703
Recio, T. II-761
Redondo, Miguel A. III-786
Reitsma, Femke II-1069
Remigi, Andrea III-745
Remolar, I. II-703
Rho, SeungMin IV-859
Ribagorda, Arturo I-812
Riganelli, Antonio II-374, II-827
Rivera-Campo, Eduardo III-22
Ro, Yong Man III-602
Robinson, Andrew III-443
Robles, Víctor II-87
Rodionov, Alexey S. III-315, IV-431
Rodionova, Olga K. III-315, IV-431
Rodríguez O., Guillermo III-415, IV-77
Rodríguez, Judith II-922
Rogerson, Peter II-1096
Roh, Sun-Sik I-1035
Roh, Yong-Wan I-89
Rosi, Marzio II-412
Rotger, Llorenç Huguet i I-924
Roy, Sasanka III-42
Rui, Zhao IV-127
Ruskin, Heather J. III-473, III-498
Rutigliano, M. II-366
Ryoo, Intae I-1026
Ryou, Hwang-bin I-1060
Ryou, Jaecheol I-776
Ryu, Eun-Kyung IV-603, IV-655, IV-665, IV-672
Ryu, So-Hyun I-319
Ryu, Tae W. IV-185, IV-791

Safouhi, Hassan II-280
Samavati, Faramarz F. III-237, III-247
Sampaio, Alcínia Zita II-817
Sánchez, Alberto II-87
Sánchez, Carlos II-328
Sánchez, Ricardo II-603
Sánchez, Teresa I-949
Sanna, N. II-366
Santaolaya Salgado, René IV-534, IV-808
Santos, Juan II-922
Santucci, A. II-1107
Sanvicente-Sánchez, Héctor III-755
Sasahara, Shinji III-11
Sastrón, Francisco III-857
Schoier, Gabriella II-1009, II-1089
Schug, Alexander III-454
Sellarès, Joan Antoni III-99
Senger, Hermes II-168
Seo, Dae-Hee I-557, III-1020
Seo, Heekyung III-837
Seo, Kyong Sok I-655
Seo, Seung-Hyun IV-689
Seo, Sung Jin I-1
Seo, Young Ro IV-964
Seong, Yeong Kyeong IV-338
Seri, Raffaello III-298
Seung-Hak, Rhee IV-948
Seznec, Andre I-960
Sgamellotti, Antonio II-412
Shahdin, S. II-350
Shen, Liran IV-414
Shen, Weidong IV-1
Shim, Hye-jin IV-321
Shim, Jae-sun IV-725
Shim, Jeong Min IV-869

Shim, Young-Chul I-1105
Shin, Byung-Joo IV-763, IV-849
Shin, Dong-Ryeol I-434
Shin, Hayong II-583
Shin, Ho-Jun I-625
Shin, Jeong-Hoon IV-754
Shin, Seung-won I-988
Shin, Yongtae I-328
Shindin, Sergey K. III-345
Sierra, José María I-851, I-812, I-960
Silva, Fabrício A.B. da II-168
Silva, Tamer Américo da I-868
Sim, Sang Gyoo I-442
Singh, Gujit II-246
Sipos, Gergely II-37
Skala, Václav III-81, III-325
Skouteris, Dimitris II-357
Slim, Chokri III-935
Smith, William R. II-392
So, Won-Ho IV-994
Sodhy, Gian Chand IV-147
Sohn, Sungwon I-776
Sohn, Won-Sung IV-743, IV-772
Sohn, Young-Ho IV-441
Song, Geun-Sil I-159, I-722
Song, Hyoung-Kyu I-386, I-394
Song, Il Gyu I-792
Song, Jin-Young II-799
Song, Kyu-Yeop IV-994
Song, Mingli III-886, IV-406
Song, Myunghyun I-294
Song, Seok Il IV-869
Song, Sung Keun IV-627
Song, Teuk-Seob IV-743
Sosa, Víctor J. Sosa IV-137
Soto, Leonardo II-603
Sousa Jr., Rafael T. de I-868
Soykan, Gürkan III-795
Stefano, Marco Di II-412
Stehlík, J. II-456
Stevens-Navarro, Enrique IV-177
Stögner, Herbert III-538
Strandbergh, Johan I-821
Studer, Pedro II-817
Sturm, Patrick III-109
Sug, Hyontai IV-158
Sugihara, Kokichi III-53, III-71, III-227
Sulaiman, Md Nasir II-146
Sun, Jizhou II-252, II-272

Tae, Kang Soo I-114
Talia, Domenico II-55
Tan, Rebecca B.N. II-873
Tang, Chuan Yi III-355
Taniar, David III-508, IV-543
Tasaltin, Cihat IV-398
Tasso, Sergio II-437
Tavadyan, Levon A. II-313
Techapichetvanich, Kesaraporn IV-479
Tejel, Javier III-22
Temurtas, Fevzullah IV-389, IV-398
Temurtas, Hasan IV-398
Thanh, Nguyen N. III-602
Thulasiram, Ruppa K. III-686
Thulasiraman, Parimala III-686
Togores, Reinaldo II-641, II-779, III-158
Tomás, Ana Paula III-117, III-127
Tomascak, Andrew III-90
Torres, Joaquín I-851
Torres-Jimenez, Jose IV-506
Trendafilov, Nickolay T. II-952
Turányi, Tamás II-226

Uhl, Andreas III-538
Uhmn, Saangyong I-81
Um, Sungmin I-57

Valdés Marrero, Manuel A. IV-137, IV-534, IV-808
Vanmechelen, Kurt IV-514
Vanzi, Eleonora II-495
Varnuška, Michal II-682
Vásquez Mendez, Isaac M. IV-534, IV-808
Vavřík, P. II-456
Vehreschild, Andre II-882
Ventura, Immaculada III-207
Verduzco Medina, Francisco IV-137
Ves, Esther De IV-887
Villalba, Luis Javier García I-859, I-868
Voloshin, V.P. III-217

Wang, Huiqiang IV-414
Wang, Tong III-706
Wang, Xiaolin III-335
Watson, Anthony IV-471
Wenzel, Wolfgang III-454, III-465
Willatzen, Morten III-817

Winter, S.C. II-30
Won, Dongho I-645, I-895, III-955
Woo, Yoseop I-270, I-345, III-585
Wouters, Carlo III-508
Wozniak, Michal III-593
Wu, Bang Ye III-355
Wu, Guohua II-731
Wyvill, Brian III-247

Xinyu, Yang IV-127
Xu, Guang III-277
Xu, Jinhui III-277
Xu, Qing II-693
Xu, Zhuoqun III-335

Yamada, Ikuho II-1096
Yan, Shaur-Uei II-721
Yang, Bailin II-236
Yang, Jin S. I-191, I-693, IV-681
Yang, Jong-Un IV-359
Yang, Shulin IV-1
Yang, Sun Ok I-286
Yang, Sung-Bong II-741, IV-743
Yang, SunWoong I-73
Yang, Tz-Hsien II-713
Yang, Zhiling II-126
Yao, Zhenhua III-729
Yap, Chee III-62
Yaşar, Osman III-795, III-807
Yavari, Issa II-432
Yen, Sung-Ming I-150
Yi, Myung-Kyu III-945, IV-584
Yi, Shi IV-127
Yim, Wha Young IV-964
Yin, Xuesong II-731
Yoe, Hyun I-294, I-402
Yong, Chan Huah IV-147
Yoo, Hyeong Seon I-510
Yoo, Jae Soo IV-869
Yoo, Kee-Young III-638, IV-87, IV-196, IV-603, IV-617, IV-655, IV-665, IV-672
Yoo, Kil-Sang I-159
Yoo, Kook-yeol IV-329
Yoo, Sang Bong II-1079
Yoo, Weon-Hee IV-524
Yoo, Wi Hyun IV-196
Yoon, Eun-Jun IV-665
Yoon, Hyung-Wook IV-46
Yoon, Jin-Sung I-9
Yoon, Ki Song II-46
Yoon, Miyoun I-328
You, Il-Sun I-167
You, Mingyu III-886, IV-406
You, Young-Hwan I-386, I-394
Youn, Chan-Hyun I-352
Youn, Hee Yong I-114, I-711, III-1010, IV-627, IV-637
Yu, Chansu IV-97
Yu, Kwangseok III-62
Yu, Qizhi II-236
Yum, Dae Hyun I-471, I-802
Yumusak, Nejat IV-389, IV-398
Yun, Byeong-Soo IV-818
Yun, Miso II-892

Zaia, Annamaria III-575
Zeng, Qinghuai II-158
Zhang, Hu III-764
Zhang, Jiawan II-252, II-272, II-693
Zhang, Jing IV-994
Zhang, Mingmin II-236
Zhang, Minming III-308
Zhang, Qin II-116
Zhang, Rubo IV-414
Zhang, Yi II-272
Zhang, Zhaoyang IV-243
Zhao, Chunjiang II-751
Zhou, Jianying I-903
Zhu, Binhai III-90, III-257
Zotta, D. II-1046