Lecture Notes in Computer Science 5370

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

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Alfred Kobsa
University of California, Irvine, CA, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

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
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Lishan Kang Zhihua Cai Xuesong Yan
Yong Liu (Eds.)

Advances in Computation and Intelligence

Third International Symposium, ISICA 2008
Wuhan, China, December 19-21, 2008
Proceedings

Springer

Volume Editors

Lishan Kang
Zhihua Cai
Xuesong Yan
China University of Geosciences
School of Computer Science
Wuhan, Hubei 430074, China
E-mail: kang_whu@yahoo.com, zhcai@cug.edu.cn, yanxs1999@hotmail.com

Yong Liu
The University of Aizu
Tsuruga, Ikki-machi, Aizu-Wakamatsu, Fukushima 965-8580, Japan
E-mail: yliu@u-aizu.ac.jp

Library of Congress Control Number: 2008940440

CR Subject Classification (1998): C.1.3, I.2, I.2.6, I.5.1, F.1, H.2.8, J.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-92136-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-92136-3 Springer 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. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2008
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12581115 06/3180 5 4 3 2 1 0

Preface

We are proud to introduce the proceedings of the Third International Symposium on Intelligence Computation and Applications (ISICA 2008) held at the China University of Geosciences (Wuhan), China, during December 19–21, 2008. ISICA 2008 successfully attracted nearly 700 submissions. Through rigorous reviews, 93 high-quality papers were included in the proceedings of ISICA 2008. ISICA conferences are one of the first series of international conferences on computational intelligence that combine elements of learning, adaptation, evolution and fuzzy logic to create programs as alternative solutions to artificial intelligence. The proceedings of ISICA conferences have a number of special features including uniqueness, novelty, success, and broadness of scope. The proceedings of ISICA conferences have been accepted in the Index to Scientific and Technical Proceedings (ISTP), while the ISICA 2007 proceedings have also been indexed by Engineering Information (EI).

Following the success of ISICA 2005 and ISICA 2007, ISICA 2008 made good progress on analyzing and processing massive real-time data by computational intelligence. ISICA 2008 featured the most up-to-date research in computational intelligence, evolutionary computation, evolutionary multi-objective and dynamic optimization, evolutionary learning systems, neural networks, classification and recognition, bioinformatics and bioengineering, evolutionary data mining and knowledge discovery, intelligent GIS and control, theory of intelligent computation, combinational and numerical optimization, and real-world applications. ISICA 2008 provided a venue to foster technical exchanges, renew everlasting friendships, and establish new connections.

On behalf of the Organizing Committee, we would like to thank warmly the sponsors, China University of Geosciences, who helped in one way or another to achieve our goals for the conference. We wish to express our appreciation to Springer for publishing the proceedings of ISICA 2008. We also wish to acknowledge the dedication and commitment of the LNCS editorial staff. We would like to thank the authors for submitting their work, as well as the Program Committee members and reviewers for their enthusiasm, time and expertise. The invaluable help of active members from the Organizing Committee, including Wenyin Gong, Xiaobo Liu, Bo Huang, Ying Chen, Jiewei Li, Pang Fang, and Pengfei Jiang, in setting up and maintaining the online submission systems, assigning the papers to the reviewers, and preparing the camera-ready version of the proceedings is highly appreciated. We would like to thank them personally for helping make ISICA 2008 a success.

December 2008

Lishan Kang
Zhihua Cai
Xuesong Yan
Yong Liu

Organization

ISICA 2008 was organized by the School of Computer Science, China University of Geosciences, and sponsored by China University of Geosciences.

General Chair

Yanxin Wang	China University of Geosciences, China

Program Chair

Lishan Kang	China University of Geosciences, China

General Co-chair

Zhihua Cai	China University of Geosciences, China
Yong Liu	The University of Aizu, Japan

Program Co-chair

Charles X. Ling	The University of Western Ontario, Canada

Program Committee

Hussein A. Abbass	University of New South Wales, Australia
Tughrul Arslan	The University of Edinburgh, UK
Wolfgang Banzhaf	Memorial University of Newfoundland, Canada
Zhihua Cai	China University of Geosciences, China
Guoliang Chen	University of Science and Technology of China, China
Ying-Ping Chen	National Chiao Tung University, Taiwan, China
Carlos A. Coello	LANIA, Mexico
Guangming Dai	China University of Geosciences, China
Kalyanmoy Deb	Indian Institute of Technology, India
Lixin Ding	Wuhan University, China
Candida Ferreira	Gepsoft
Garry Greenwood	Portland State University, Portland
Jun He	University of Birmingham, UK
Xingui He	Peking University, the Chinese Academy of Engineering, China
Zhenya He	Eastsouth University, the Chinese Academy of Sciences, China

Tetsuya Higuchi	National Institute of Advanced Industrial Science and Technology, Japan
Zhangcan Huang	Wuhan University of Technology, China
Hisao Ishibuchi	Osaka Prefecture University, Japan
Licheng Jiao	Xidian University, China
Lishan Kang	China University of Geosciences, China
John R. Koza	Stanford University, USA
Lawrence W. Lan	National Chiao Tung University, Taiwan, China
Yuanxiang Li	Wuhan University, China
Zhenhua Li	China University of Geosciences, China
Guangxi Liang	Chinese University of Hong Kong, China
Jiajun Lin	East China University of Science and Technology, China
Yong Liu	The University of Aizu, Japan
Bob Mckay	Seoul National University, South Korea
Zbigniew Michalewicz	University of North Carolina, USA
Erkki Oja	University of Technology Helsinki, Finland
Ping-Feng Pai	National Chi Nan University, Taiwan, China
Peter Ross	Napier University, UK
Wei-Chiang Samuelson Hong	Oriental Institute of Technology, Taiwan, China
Marc Schoenauer	Université Paris Sud, France
He Shen	University of Birmingham, UK
Zhongzhi Shi	Institute of Computing Technology, China
Hsu-Shih Shih	Tamkang University, Taiwan, China
Dianxun Shuai	East China University of Science Technology, China
Huai-Kuang Tsai	Institute of Information Science, Academia Sinica, Taiwan, China
Edward Tsang	University of Essex, UK
Shaowei Wang	Nanjing University, China
Zhijian Wu	Wuhan University, China
Tao Xie	National University of Defense Technology, China
Shenwu Xiong	Wuhan University of Technology, China
Zongben Xu	Xi'an Jiaotong University, China
Shengxiang Yang	University of Leicester, UK
Yu Yang	Nanjing University, China
Gary G. Yen	Oklahoma State University, USA
Lotfi A. Zadeh	University of California Berkeley, USA
Jianchao Zeng	Taiyuan University of Technology, China
Sanyou Zeng	China University of Geosciences, China
Ba Zhang	Tsinghua University, the Chinese Academy of Sciences, China
Huajie Zhang	University of New Brunswick, Canada

Qingfu Zhang	University of Essex, UK
Jinhua Zheng	Xiangtan University, China
Xiufen Zou	Wuhan University, China

Local Chair

Yadong Liu	China University of Geosciences, China

Local Co-chair

Guangming Dai	China University of Geosciences, China
Hui Li	China University of Geosciences, China
Zhenhua Li	China University of Geosciences, China
Sifa Zhang	China University of Geosciences, China

Local Committee

Shuanghai Hu	China University of Geosciences, China
Liangxiao Jiang	China University of Geosciences, China
Siwei Jiang	China University of Geosciences, China
Xiang Li	China University of Geosciences, China
Siqing Xue	China University of Geosciences, China
Xuesong Yan	China University of Geosciences, China
Huili Zhang	China University of Geosciences, China

Secretaries

Ying Chen	China University of Geosciences, China
Pan Fang	China University of Geosciences, China
Wenyin Gong	China University of Geosciences, China
Bo Huang	China University of Geosciences, China
Pengfei Jiang	China University of Geosciences, China
Jiewei Li	China University of Geosciences, China
Xiaobo Liu	China University of Geosciences, China

Sponsoring Institutions

China University of Geosciences, Wuhan, China

Table of Contents

Section I: Computational Intelligence

A Balanced Ensemble Learning with Adaptive Error Functions 1
Yong Liu

A Self-adaptive Evolutionary Programming Based on Optimum Search Direction 9
Guangming Lin, Xin Lu, Yongsheng Liang, Lishan Kang, and Xin Yao

Dynamic Clonal and Chaos-Mutation Evolutionary Algorithm for Function Optimization 19
Ming Yang and Jing Guan

A Parallel Self-adaptive Subspace Searching Algorithm for Solving Dynamic Function Optimization Problems 28
Yan Li, Zhuo Kang, and Lishan Kang

Improved GuoTao Algorithm for Unconstrained Optimization Problems 37
Ziyi Chen, Lishan Kang, and Lijun Liu

Gene Expression Programming Based on Subexpression Library and Clonal Selection 45
Siqing Xue and Jie Wu

Correlation between Mutations and Self-adaptation in Evolutionary Programming 58
Yong Liu

Representations of Evolutionary Electronics 67
Xuesong Yan, Pan Fang, Qingzhong Liang, and Chenyu Hu

Section II: Evolutionary Computation

A Novel Particle Swarm Optimization for Constrained Engineering Optimization Problems 79
Minghai Jiao and Jiafu Tang

An Improved Differential Evolution Based on Triple Evolutionary Strategy 89
Yichao He, Yingzhan Kou, and Chunpu Shen

Cluster-Degree Analysis and Velocity Compensation Strategy of PSO ... 98
Quansheng Dou, Zhijun Yu, Zhongzhi Shi, Erkeng Yu, and Yongzhi Zheng

Combine LHS with MOEA to Optimize Complex Pareto Set MOPs 107
Jinhua Zheng, Biao Luo, Miqing Li, and Jing Li

Energy-Predicted Shortest Routing Tree Algorithm in Wireless Sensor Networks ... 117
Ming Zhang, Chenglong Gong, Yuan Feng, and Chao Liu

Evolutionary Antenna Design via Modified Normalized GT Algorithm ... 125
Yuanyuan Fan, Qingzhong Liang, Zhenhua Cai, and Hui Li

Quantitative Cryptanalysis of Six-Round DES Using Evolutionary Algorithms ... 134
Fan Yang, Jun Song, and Huanguo Zhang

Section III: Evolutionary Multi-objective and Dynamic Optimization

A K-Nearest-Neighbors Pareto Rank Assignment Strategy and Compound Crossover Operator Based NSGA-II and Its Applications on Multi-objective Optimization Functions ... 142
Weiya Guo, Zhenhua Li, Dan Zhao, and Tim Wong

A Multi-objective Differential Evolutionary Algorithm Based on Spacial Distance ... 152
Jinhua Zheng, Jun Wu, and Hui Lv

A Novel Opposition-Based Multi-objective Differential Evolution Algorithm for Multi-objective Optimization ... 162
Lei Peng, Yuanzhen Wang, and Guangming Dai

Acquainted Non-convexity Multiresolution Based Optimization for Affine Parameter Estimation in Image Registration ... 171
J. Dinesh Peter, V.K. Govindan, and Abraham T. Mathew

An Improved Hybrid Multi-objective Particle Swarm Optimization Algorithm ... 181
Zuan Zhou, Guangming Dai, Pan Fang, Fangjie Chen, and Yi Tan

Application of Ant Colony Optimization Algorithm to Multi-Join Query Optimization ... 189
Nana Li, Yujuan Liu, Yongfeng Dong, and Junhua Gu

A Hybrid Algorithm for Vehicle Routing Problem with Time Windows 198
Dengying Jiang, Wenxia Jiang, and Zhangcan Huang

Section IV: Evolutionary Learning Systems

An Architecture Design for Evolvable Computer 206
Xuesong Yan, Chen Shi, Lishan Kang, and Shitan Huang

Gene Expression Programming Neural Network for Regression and Classification 212
Weihong Wang, Qu Li, and Xing Qi

Granular-Based Linguistic Models for Identification of Process System 220
Keun-Chang Kwak

Particle Swarm Optimization Applied to Restoration of Electrical Energy Distribution Systems 228
Germano Lambert-Torres, Helga Gonzaga Martins, Maurilio Pereira Coutinho, Camila Paes Salomon, and Leonardo Schilling Filgueiras

Space Regression Analysis on Geochemical Data by the GEP Evolutionary Model Based on Kriging 239
Dongmei Zhang, Ao Wang, and Zhifen Chen

Fault Tolerance Improvement through Architecture Change in Artificial Neural Networks 248
Fernando Morgado Dias and Ana Antunes

A Strategic Model for Measuring Agility with Fuzzy Logic 258
Gholamreza Khoshsima

Section V: Neural Networks

Comparison on Gradient-Based Neural Dynamics and Zhang Neural Dynamics for Online Solution of Nonlinear Equations 269
Yunong Zhang, Chenfu Yi, and Weimu Ma

Competitive Hopfield Neural Network with Periodic Stochastic Dynamics for Partitional Clustering 280
Zhanghui Kuang, Wei Bi, and Jiahai Wang

Modeling Hysteresis in Piezo Actuator Based on Neural Networks 290
Xuefeng Yang, Wei Li, Yuqiao Wang, and Guo Ye

Modeling of Grey Neural Network and Its Applications 297
Jingling Yuan, Luo Zhong, Xiaoyan Li, and Jie Li

On-Line Estimation of Biomass Concentration Based on ANN and Fuzzy C-Means Clustering 306
Guohai Liu, Haixia Xu, Dawei Zhou, and Congli Mei

PCA-Based Elman Neural Network Algorithm 315
Shifei Ding, Weikuan Jia, Chunyang Su, Xinzheng Xu, and Liwen Zhang

Prediction Model of Water Resources in Mine Area Based on Phase Space Reconstruction and Chaos Neural Network 322
Keping Zhou, Ge Gao, Feng Gao, and Wenxiang Gao

Predictive Control Strategy of Hydraulic Turbine Turning System Based on BGNN Neural Network 331
Yijian Liu and Yanjun Fang

The Application of Improved BP Neural Network Algorithm in Lithology Recognition 342
Yuxiang Shao, Qing Chen, and Dongmei Zhang

Section VI: Classification and Recognition

A Combined Classification Algorithm Based on C4.5 and NB 350
Liangxiao Jiang, Chaoqun Li, Jia Wu, and Jian Zhu

A Keyword Extraction Method Based on Lexical Chains 360
Xiao-yu Jiang

A Method of the Extraction of Texture Feature 368
Haifang Li, Lihuan Men, and Junjie Chen

A Transductive Learning Method for Interactive Image Segmentation ... 378
Jiazhen Xu, Xinmeng Chen, Yang Wei, and Xuejuan Huang

A New Approach of Feature Selection for Chinese Web Page Categorization 386
Cunhe Li, Lina Zhu, and Kangwei Liu

Easily Reconfigurable Analytical Fuzzy Predictive Controllers: Actuator Faults Handling 396
Piotr M. Marusak

Incremental Manifold Learning Algorithm Using PCA on Overlapping Local Neighborhoods for Dimensionality Reduction 406
Yubin Zhan, Jianping Yin, Guomin Zhang, and En Zhu

Intelligent 3D Face Recognition 416
Chao Li

Language Feature Mining for Music Emotion Classification via Supervised Learning from Lyrics 426
Hui He, Jianming Jin, Yuhong Xiong, Bo Chen, Wu Sun, and Ling Zhao

Section VII: Bioinformatics and Bioengineering

An Immune Multi-agent System for Network Intrusion Detection 436
Dian Gang Wang, Tao Li, Sun Jun Liu, Gang Liang, and Kui Zhao

A Novel Biology-Inspired Virus Detection Model with RVNS 446
Renchao Qin, Tao Li, and Yu Zhang

A Novel Exponential Type Swarming of Foraging and Obstacle-Avoidance Behaviour Modelling and Simulating Research on Collective Motion in Multi-obstacle Environment 454
Zhi Bin Xue and Jian Chao Zeng

Algorithm of On-Line Handwriting Signature Verification Based on Discrete Fréchet Distance 461
Jianbin Zheng, Xiaolei Gao, Enqi Zhan, and Zhangcan Huang

Dynamic Model and Numerical Simulation of Particle Motion in Rotation Flow Field of Centrifuge 470
Tong Zhu, Tsutomu Nozaki, Yuanhua Xie, Jin Han, Jing Jiang, and Bing Li

Impact on Genetic Algorithm of Different Parameters 479
Xiang Li, Qiao Chen, and Yanli Li

New Fast Decision Tree Classifier for Identifying Protein Coding Regions 489
Hazem M. El-Bakry and Mohamed Hamada

Phase Transition in the Evolution of Artificial Life on Random Networks 501
Wei Qiang, Hui Li, and Hui Cao

Section VIII: Evolutionary Data Mining and Knowledge Discovery

A Novel Automatic Parameters Optimization Approach Based on Differential Evolution for Support Vector Regression 510
Jiewei Li and Zhihua Cai

An Improved Gene Expression Programming for Fuzzy Classification . . . 520
Xiaobo Liu, Zhihua Cai, and Wenyin Gong

An Improved K-Means Clustering Algorithm Based on Spectral Method 530
Shengwen Tian, Hongyong Yang, Yilei Wang, and Ali Li

E-Means: An Evolutionary Clustering Algorithm 537
Wei Lu, Hengjian Tong, and Issa Traore

Finding Motifs of Financial Data Streams in Real Time 546
Tao Jiang, Yucai Feng, Bin Zhang, Jie Shi, and Yuanzhen Wang

Improve the Accuracy of One Dependence Augmented Naive Bayes by Weighted Attribute 556
Siwei Jiang and Zhihua Cai

Multi-Relational Classification in Imbalanced Domains 562
Guangmei Xu, Hong Bao, and Xianyu Meng

Sandstorm Occurrence Frequency Short-Term Prediction Based on Bootstrap Method 571
Jianli Dong, Haiyang Chen, Jianzhou Wang, and Donghuai Sun

Simplify Multi-valued Decision Trees 581
Chien-Liang Liu and Chia-Hoang Lee

Section IX: Intelligent GIS and Control

A GIS-Based Evaluation on Sensitivity of Soil Erosion in Yishusi River Watershed 591
Jia Yan, Jun Du, Chongsheng Xue, and Qinghua Yang

A New Universal Combinatorial Operation Model with Unequal Weights 599
Lihua Fu and Huacan He

An Observer-Based Neural Network Controller for Chaotic Lorenz System 608
Suwat Kuntanapreeda

Integrated Agent-Based Modeling with GIS for Large Scale Emergency Simulation 618
Dan Guo, Bo Ren, and Cheng Wang

New Research on Harmonics and Reactive Currents Detecting and Its Suppression with ANN in Single-Phase Circuit 626
Wenjin Dai, Qingsheng Lin, and Yongtao Dai

Qualitative Spatio-temporal Reasoning about Moving Objects in Three-Dimensional Space 637
Jingde Cheng

The Design of LQR Controller Based on Independent Mode Space for Active Vibration Control 649
Jingjun Zhang, Lili He, Ercheng Wang, and Ruizhen Gao

Section X: Theory of Intelligent Computation

A Multiagent Genetic Particle Swarm Optimization 659
Lianguo Wang, Yi Hong, Fuqing Zhao, and Dongmei Yu

About the Computation Time of Adaptive Evolutionary Algorithms 669
Lixin Ding and Jinghu Yu

Frequency-Domain Adaptive Filtering of Batch Dealing Used for GPS Anti-jamming 680
Hongtao Song, Lin Zhao, Jicheng Ding, and Ming Sun

Image Decomposition Based on Curvelet and Wave Atom 687
Chengwu Lu

Theoretical and Empirical Analysis of the Learning Rate and Momentum Factor in Neural Network Modeling for Stock Prediction . . . 697
Jinchuan Ke, Xinzhe Liu, and Guan Wang

Exploring Building Blocks through Crossover 707
Zhenhua Li and Erik D. Goodman

Section XI: Combinational and Numerical Optimization

A Chaotic Neural Network Combined Heuristic Strategy for Multidimensional Knapsack Problem 715
Ying Zhou, Zhanghui Kuang, and Jiahai Wang

A New Optimization Algorithm for Weight Optimization 723
Hui Li and Xuesong Yan

Combination of Global and Local Search for Real Function Optimization 731
Xinsheng Lai

Multi-Classifier Systems (MCSs) of Remote Sensing Imagery Classification Based on Texture Analysis 740
Hongfen Li, Guangdao Hu, and Jiang-feng Li

Particle Swarm Optimization with Dynamic Dimension Crossover for High Dimensional Problems 750
Chengyu Hu, Xuesong Yan, and Chuanfeng Li

Stability Analysis of Discrete Hopfield Neural Networks with Weight Function Matrix 760
Jun Li, Yongfeng Diao, Jiali Mao, Ying Zhang, and Xing Yin

Hybrid Ant Colony Algorithm and Its Application on Function Optimization 769
Bo Liu, Huiguang Li, Tihua Wu, and Qingbin Zhang

Section XII: Real-World Applications

Adaptive Shot Change Detection for Hardware Application 778
WonHee Kim, KwangSeok Moon, and JongNam Kim

An Efficient Self Routing Scheme by Using Parent-Child Association for WSNs 785
Yeon-Mo Yang and IlSoo Jeon

Cryptanalysis of Transposition Cipher Using Simulated Annealing Genetic Algorithm 795
Jun Song, Fan Yang, Maocai Wang, and Huanguo Zhang

Driver Recognition Using Gaussian Mixture Models and Decision Fusion Techniques 803
Kristin S. Benli, Remzi Düzağaç, and M. Taner Eskil

Fast Partial Distortion Elimination Algorithm Using Probability Improvement of Sub-block Uniform Distribution 812
TaeKyung Ryu, KwangSeok Moon, and JongNam Kim

PSO Combined with ILS for Flowshop-Based Parallel Task Assignment Problem 821
Runpeng Liang, Jiaxiang Luo, Qingqiang Yang, and Wenfeng Luo

Target Localization for Autonomous Soccer Robot Based on Vision Perception 831
Guifang Shao, Yuhua Wen, Fang Yu, and Zushu Li

Urban Land Classification Research Based on Data Field 841
Yu Deng, Haijun Wang, Li Wang, and Dandan Shan

Research and Design of Digital Synthesizer Based on MATLAB 849
Chunjing Mao, Yong Guan, and Yongmei Liu

Author Index 859

A Balanced Ensemble Learning with Adaptive Error Functions

Yong Liu

The University of Aizu
Aizu-Wakamatsu, Fukushima 965-8580, Japan
yliu@u-aizu.ac.jp

Abstract. In the practice of designing neural network ensembles, it is common that a certain learning error function is defined and kept the same or fixed for each individual neural network in the whole learning process. Such fixed learning error function not only likely leads to over-fitting, but also makes learning slow on hard-learned data points in the data set. This paper presents a novel balanced ensemble learning approach that could make learning fast and robust. The idea of balanced ensemble learning is to define adaptive learning error functions for different individual neural networks in an ensemble, in which different individuals could have different formats of error functions in the learning process, and these error functions could be changed as well. Through shifting away from well-learned data and focusing on not-yet-learned data by changing error functions for each individual among the ensemble, a good balanced learning could be achieved for the learned ensemble.

1 Introduction

According to the principle of divide and conquer, a complex computational task is solved by dividing it into a number of computationally simple tasks and then combining the solutions to those tasks. In machine learning, such principle has led to the development of ensemble systems that consist of several subsystems being able to reduce the total complexity of the system while solving a difficult problem satisfactorily. Basically, an ensemble system fuses knowledge learned by each subsystem to arrive at an overall decision that is supposedly superior to that reached by any one subsystem acting alone. A neural network ensemble is an example of ensemble systems in which each subsystem is a single neural network. With the advantages of neural network ensembles and the growing complexity of the real-world problems in applications, neural network ensembles have emerged as one of the most important and powerful problem-solving techniques in the last decade.

Since 1990's, a number of ensemble learning approaches have been developed from a variety of backgrounds. They could fall into three categories: independent ensemble learning, sequential ensemble learning, and simultaneous ensemble learning [1,2,3,4,5,6,7]. One example of independent ensemble learning is

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 1–8, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Breiman's bagging [3] where each learner is provided with a set of patterns obtained randomly from re-sampling the original set of examples, and then trained independently of the other learners. Freund and Schapire's boosting [4] is a typical example of sequential ensemble learning where a set of learners are trained sequentially on data sets with entirely different distributions. Liu's negative correlation learning is a representative of simultaneous ensemble learning where all individual neural networks are trained simultaneously and interactively [7].

Although these developed ensemble learning approaches have been shown to perform well in practice, a formal framework for ensemble learning is still in its infancy. As a result, these ensemble learning approaches suffer from some deficiencies such as its being sensitive to the choice of learning error functions and parameters, lacking of generalization error bound, and few results on their scaling to large problems. Consequently, ensemble learning is of limited use to general practitioners.

This paper proposed a novel balanced ensemble learning approach for neural network ensembles in order to achieve robust performance without being affected by the choices of learning functions. The idea of balanced ensemble learning is to introduce adaptive learning error functions for different individual neural networks in an ensemble, in which different individuals could have different formats of error functions in the learning process, and these error functions could be changed as well. It is different from previous work in ensemble learning where error functions are often set to the same or the fixed ones in the whole learning process. Generally speaking, on learning data where the ensemble has learned well, the error functions in balanced ensemble learning would be changed to allow the ensemble to have a little larger values in error so that the ensemble could shift its attention of learning away from these well-learned data. On other learning data where the ensemble has not yet learned, enlarged error signals are given to each individual so that the ensemble could focus on these not-yet-learned data. Through shifting away from well-learned data and focusing on not-yet-learned data, a good balanced learning could be achieved in the ensemble. It is also the reason that ensemble learning with adaptive error functions proposed in this paper is called balanced ensemble learning between the shifting process and the enforcing process.

The rest of this paper is organized as follows: Section 2 describes ideas of balanced ensemble learning. Section 3 compared results between balanced ensemble learning and negative correlation learning on four real world problem data sets, and discusses when and how balanced ensemble learning works. Finally, Section 4 concludes with a summary of the paper.

2 Balanced Ensemble Learning

A balanced ensemble learning can be developed by changing error functions in negative correlation learning (NCL). In NCL, the output y of a neural network ensemble is formed by a simple averaging of outputs F_i of a set of neural networks. Given the training data set $D = \{(\mathbf{x}(1), y(1)), \cdots, (\mathbf{x}(N), y(N))\}$, all the individual networks in the ensemble are trained on the same training data set D

$$F(n) = \frac{1}{M}\Sigma_{i=1}^{M} F_i(n) \tag{1}$$

where $F_i(n)$ is the output of individual network i on the nth training pattern $\mathbf{x}(n)$, $F(n)$ is the output of the neural network ensemble on the nth training pattern, and M is the number of individual networks in the neural network ensemble.

The idea of NCL [7] is to introduce a correlation penalty term into the error function of each individual network so that all the individual networks can be trained simultaneously and interactively. The error function E_i for individual i on the training data set D in negative correlation learning is defined by

$$E_i = \frac{1}{N}\Sigma_{n=1}^{N} E_i(n) = \frac{1}{N}\Sigma_{n=1}^{N}\left[\frac{1}{2}(F_i(n) - y(n))^2 - \lambda\frac{1}{2}(F_i(n) - F(n))^2\right] \tag{2}$$

where N is the number of training patterns, $E_i(n)$ is the value of the error function of network i at presentation of the nth training pattern, and $y(n)$ is the desired output of the nth training pattern. The first term in the right side of Eq.(2) is the mean-squared error of individual network i. The second term is a correlation penalty function. The purpose of minimizing is to negatively correlate each individual's error with errors for the rest of the ensemble. The parameter λ is used to adjust the strength of the penalty.

The partial derivative of E_i with respect to the output of individual i on the nth training pattern is

$$\begin{aligned}\frac{\partial E_i(n)}{\partial F_i(n)} &= F_i(n) - y(n) - \lambda(F_i(n) - F(n)) \\ &= (1-\lambda)(F_i(n) - y(n)) + \lambda(F(n) - y(n))\end{aligned} \tag{3}$$

In the case of $0 < \lambda < 1$, both $F(n)$ and $F_i(n)$ are trained to go closer to the target output $y(n)$ by NCL. $\lambda = 0$ and $\lambda = 1$ are the two special cases. At $\lambda = 0$, there is no correlation penalty function, and each individual network is just trained independently based on

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F_i(n) - y(n) \tag{4}$$

At $\lambda = 1$, the derivative of error function is given by

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F(n) - y(n) \tag{5}$$

where the error signal is decided by $F(n) - y(n)$, i.e. the difference between $F(n)$ and $y(n)$.

For the classification problems, it is unnecessary to have the smallest difference between $F(n)$ and $y(n)$, but to have enough difference between $F(n)$ and $y(n)$ in order to have the correct decision. For a two-class problem, the target value y on a data point can be set up to 1.0 or 0.0 depend on which class the data

point belongs to. As long as F is larger than 0.5 at $y = 1.0$ or smaller than 0.5 at $y = 1.0$, the data point will be correctly classified.

In balanced ensemble learning, the error function for each individual on each data point is defined based on whether the ensemble learned the data point or not. If the ensemble had learned to classify the data point correctly, a shifting parameter β could be introduced into the derivative of error function in Eq.(refrelation) for each individual

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F(n) - |y(n) - \beta| \tag{6}$$

Otherwise, an enforcing parameter α would be added to the the derivative of error function for each individual

$$\frac{\partial E_i(n)}{\partial F_i(n)} = \alpha(F(n) - y(n)) \tag{7}$$

By shifting and enforcing the derivative of error function, the ensemble would not need to learn every data too well to prevent from learning hard data points too slowly.

3 Experimental Results

3.1 Four Data Sets

Four real-world problems, i.e. the Australian credit card assessment problem, the heart disease problem, the diabetes problem, and the cancer problem, were tested. The four data sets were obtained from the UCI machine learning benchmark repository. It is available by anonymous ftp at ics.uci.edu (128.195.1.1) in directory /pub/machine-learning-databases.

The Australian credit card assessment data set is to assess applications for credit cards based on a number of attributes. There are 690 cases in total. The output has two classes. The 14 attributes include 6 numeric values and 8 discrete ones, the latter having from 2 to 14 possible values.

The purpose of the heart disease data set is to predict the presence or absence of heart disease given the results of various medical tests carried out on a patient. This database contains 13 attributes, which have been extracted from a larger set of 75. The database originally contained 303 examples but 6 of these contained missing class values and so were discarded leaving 297. 27 of these were retained in case of dispute, leaving a final total of 270. There are two classes: presence and absence (of heart disease).

The diabetes data set is a two-class problem which has 500 examples of class 1 and 268 of class 2. There are 8 attributes for each example. The data set is rather difficult to classify. The so-called "class" value is really a binarised form of another attribute which is itself highly indicative of certain types of diabetes but does not have a one to one correspondence with the medical condition of being diabetic.

The breast cancer data set was originally obtained from W. H. Wolberg at the University of Wisconsin Hospitals, Madison. The purpose of the data set is to classify a tumour as either benign or malignant based on cell descriptions gathered by microscopic examination. The data set contains 9 attributes and 699 examples of which 458 are benign examples and 241 are malignant examples.

3.2 Experimental Setup

m-fold cross-validation were used to in which the parameter m was set to be 10 for the Australian credit card data set, the heart disease data set, and the breast cancer data set, and 12 for the diabetes data set, respectively. 5 runs of m-fold cross-validation had been conducted to calculate the average results. In another word, 60 runs for the diabetes data set and 50 runs for each of other data sets had been executed in estimating every average result.

The ensemble architecture used in the experiments has 50 networks. Each individual network is a feedforward network with one hidden layer. Two architectures had been tested in the experiments, in which all the individual networks are either small neural networks with one hidden node, or big neural networks with 10 hidden nodes. The number of training epochs was set to 2000.

3.3 Experiment Results of NCL

Table 1 described the results of the learned ensembles consisting of 50 small neural networks by NCL from 60 runs on the diabetes data set and 50 runs on the other three data sets. Each neural network has one hidden node only in the ensembles. The learned ensembles only had a low error rate 2% on the training set of the cancer data set. For the other three data sets, the error rates of the learned ensembles increased form 6.2% to 25.1% on the training set. As expected, NCL with $\lambda = 1$ could reach a little lower error rates on the training sets. While the error rates on the training set reflect the capability of the learned ensemble, the error rates on the testing set give the measurements of generalization. Except for the little change of the testing error rates on the cancer data set, the testing error rates had been found to decrease with the increase values of λ. It might suggest that the learned ensembles with small neural networks might be underfitting.

In order to examine whether the small neural networks had led to underfitting of the learned ensemble, the ensembles with 50 big neural networks were trained by NCL on four data sets. There are 10 hidden nodes in each individual neural

Table 1. Average of error rates of the learned ensembles with small neural networks by NCL on four data sets

	Card		Heart		Diabetes		Cancer	
	Train	Test	Train	Test	Train	Test	Train	Test
0	0.112	0.141	0.075	0.137	0.236	0.251	0.020	0.034
0.5	0.104	0.140	0.074	0.132	0.233	0.248	0.020	0.035
1	0.081	0.135	0.062	0.105	0.186	0.242	0.020	0.035

Table 2. Average of error rates of the learned ensembles with big neural networks by NCL on four data sets

	Card		Heart		Diabetes		Cancer	
	Train	Test	Train	Test	Train	Test	Train	Test
0	0.020	0.147	0.0038	0.053	0.158	0.250	0.010	0.041
0.5	0.017	0.146	0.0044	0.055	0.139	0.249	0.010	0.039
1	0.059	0.140	0.0048	0.084	0.146	0.241	0.019	0.036

network. The average results over 60 runs on the diabetes data set and 50 runs on the other three data sets are given in Table 2. It is obvious that the error rates on the training set could significantly reduced on all data sets. However, the ensembles with big neural networks only achieved better results, i.e., the lower error rates on the testing set, on the heart data set. The performance of the ensembles with large neural networks is nearly the same as that of the ensemble with small neural networks on the diabetes data set. The slightly worsen results on the card and the cancer data sets might be attributed by the overfitting from the learned ensembles with large neural networks.

Although the ensembles with large neural networks could learn better on the training set, there exist high probabilities of generating the learned ensembles with overfitting. In order to prevent from overfitting on the training data set, each individual neural network should not be too large. Since it is hard to determine how large each neural network should be, it would be preferable to make individual neural networks as small as possible. When NCL could not let the ensembles with small neural networks learn well, it would be interesting to see whether the same ensembles could be trained better by the balanced ensemble learning.

3.4 Experiment Results of Balanced Ensemble Learning

Table 3 presented the average results of error rates of the learned ensembles consisting of 50 small neural networks by balanced ensemble learning over 5 runs of m-fold cross-validation where m was set to 12 for the diabetes data set, and 10 for the rest three data sets. Balanced ensemble learning had been

Table 3. Average of error rates of the learned ensembles with small neural networks by balanced ensemble learning from 60 runs on Diabetes data set and 50 runs on the other three data sets

	Card		Heart		Diabetes		Cancer	
	Train	Test	Train	Test	Train	Test	Train	Test
0	0.081	0.135	0.062	0.105	0.186	0.242	0.020	0.035
0.1	0.067	0.139	0.044	0.079	0.170	0.240	0.017	0.036
0.2	0.053	0.138	0.024	0.068	0.142	0.245	0.015	0.037
0.3	0.027	0.141	0.011	0.055	0.110	0.243	0.011	0.036
0.4	0.015	0.140	0.004	0.041	0.124	0.250	0.012	0.039

examined under different values of shifting parameter β from 0 to 0.4, in which NCL is the special case of balanced ensemble learning with $\beta = 0$. In the results of balanced ensemble learning discussed in this section, enforing parameter α was set to 1.

As expected, balanced ensemble learning could learn faster and better, and reach lower error rate on the training set with larger shifting parameters. For example, on the card data set, the error rate was reduced from 8.1% to 1.5% on the training set with increasing from 0 to 0.4. Balanced ensemble learning had also significantly decreased the error rates on the training set for the rest three data sets. Not only had balanced ensemble learning dramatically reduced the error rates on the training set, but also maintained comparable or better generalization on the testing set. In another word, overfitting is not so severe in balanced ensemble learning that balanced ensemble learning is capable of improving performance of the learned ensemble on both the training set and the testing set.

4 Conclusions

In the current applications of neural network ensemble learning, the error function is often defined as the mean squared error between the output of each individual neural network and the target output. The format of such error function is fixed, which not only likely leads to over-fitting, but also makes learning slow on hard cases among learning data. In balanced ensemble learning, the error functions could be changed based on how well the ensemble has learned on the learning data. The use of adaptive error functions allows balanced ensemble learning to learn more on those not-yet-learned data while preventing from learning too much on those learned data. In this way, balanced learning process could become fast and robust.

Balanced ensemble learning had been tested with different shifting parameters while the enforcing parameter was set as 1. By increasing the weighting parameter, the ensembles should learn even better although the chance of overfitting might rise. Balanced ensemble learning had only been applied to the ensembles with one-hidden-node neural networks in this paper. One future research on balanced ensemble learning is to apply it to the ensembles with large neural networks on controlling overfitting.

References

1. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Trans. on Pattern Analysis and Machine Intelligence 12(10), 993–1001 (1990)
2. Sarkar, D.: Randomness in generalization ability: a source to improve it. IEEE Trans. on Neural Networks 7(3), 676–685 (1996)
3. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
4. Schapire, R.E.: The strength of weak learnability. Machine Learning 5, 197–227 (1990)

5. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3, 79–87 (1991)
6. Jacobs, R.A., Jordan, M.I., Barto, A.G.: Task decomposition through competition in a modular connectionist architecture: the what and where vision task. Cognitive Science 15, 219–250 (1991)
7. Liu, Y., Yao, X.: Simultaneous training of negatively correlated neural networks in an ensemble. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics 29(6), 716–725 (1999)

A Self-adaptive Evolutionary Programming Based on Optimum Search Direction

Guangming Lin[1], Xin Lu[1], Yongsheng Liang[1], Lishan Kang[2], and XinYao[3]

[1] Shenzhen Institute of Information Technology. Shenzhen, China
[2] School of Computer Science, China University of Geoscience, Wuhan, China
[3] School of Computer Science The University of Birmingham
Edgbaston, Birmingham B15 2TT, U.K.
gmlin@tom.com

Abstract. The Classical Evolutionary Programming (CEP) relies on Gaussian mutation, whereas Fast Evolutionary Programming (FEP) selects Cauchy distribution as the primary mutation operator, Improved Fast Evolutionary (IFEP) selects the better Gaussian and Cauchy distribution as the primary mutation operator. In this paper, we propose a self-adaptive Evolutionary Programming based on Optimum Search Direction (OSDEP) in which we introduce the current best global individual into mutation to guide individuals to converge according to the global search direction. Extensive empirical studies have been carried out to evaluate the performance of OSDEP, IFEP, FEP and CEP. From the experimental results on seven widely used test functions, we can show that OSDEP outperforms all of IFEP, FEP and CEP for all the test functions.

Keywords: Classical Evolutionary Programming (CEP), Fast EP (FEP), Improved EP (IFEP), Optimum Search Direction EP (OSDEP).

1 Introduction

Although evolutionary programming (EP) was first proposed as an artificial intelligence approach [1], it has been applied successfully to many numerical and combinatorial optimization problems [2, 3, 4, 18, 23]. Optimization using EP can be explained by two major steps:

1. Mutate the solutions in the current population, and
2. Select the next generation from the mutated and the current solutions.

These two steps can be regarded as a population-based version of the classical **generate**-and-**test** method [5], where mutation used to **generate** new solutions (offspring) and selection is used to **test** which of the newly generated solutions should survive to the next generation. Formulating EP as a special case of the generate-and-test method establishes a bridge between EP and other search algorithms, such as evolution strategies (ES), genetic algorithms (GA), simulated annealing (SA), tabu search (TS), Particle Swarm Optimization (PSO), and others, and thus facilitates cross-fertilization amongst different research areas.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 9–18, 2008.
© Springer-Verlag Berlin Heidelberg 2008

One disadvantage of EP in solving some of the multi-modal optimization problems is its slow convergence to a good near-optimum solution. The generate-and-test formulation of EP indicates that mutation is a key search operator which generates new solutions from the current ones. In [6], a new mutation operator based on Cauchy random numbers has been proposed and test on the suite of 23 functions. The new EP with Cauchy mutation significantly outperforms the classical EP (CEP), which uses Gaussian mutation, on a number of multi-modal functions with many local minima while being comparable to CEP for unimodal and multi-modal function with only a few local minima.

Extensive empirical studies of both FEP and CEP have been carried out in order to evaluate the relative strengths and weaknesses of FEP and CEP for different problems. The results show that Cauchy mutation is an efficient search operator for a large class of multi-modal function optimization problems. All the parameters used in the FEP were set similar to those used in CEP for a fair comparison. However FEP's performance may be further improved using other parameters which will be proposed in a later section.

2 Optimization by EPs

A global minimization problem can be formalized as a pair (S, f), where $S \subseteq R^n$ is a bounded set on R^n and f: $S \mapsto R$ is an n-dimensional real-valued function. The problem is to find a point $x_{min} \in S$ such that $f(x_{min})$ is a global minimum on S. More specifically, it is required to find an $x_{min} \in S$ such that

$$\forall x \in S : f(x_{\min}) \le f(x),$$

Where f does not need to be continuous but, it must be bounded. This paper only considers unconstrained function optimization.

2.1 Classical Evolutionary Programming (CEP)

Fogel [1, 3] and Bäck and Schwefel [7] have indicated that CEP with self-adaptive mutation usually performs better that CEP without self-adaptive mutation for the function they test. Hence, CEP with self-adaptive mutation will be investigated in this paper. According to the description by Bäck and Schwefel [7], in this study CEP is implemented as follows:

1. Generate the initial population of μ individuals, and set k=1. Each individual is taken as a pair of real-valued vectors (x_i, η_i), $\forall i \in \{1, 2, ..., \mu\}$, where x_i's are variables and η_i's are standard deviations for Gaussian mutations (also know as strategy parameters in self-adaptive evolutionary algorithms).
2. Evaluate the fitness score for each individual $(x_i, \eta_i), \forall i \in \{1, 2, ..., \mu\}$, of the population based on the objective function $f(x_i)$.

3. Each parent $(x_i, \eta_i), \forall i \in \{1,2,...,\mu\}$, creates a single offspring (x_i ', η_i ') by: for j=1,2,…n,

$$x_i'(j) = x_i(j) + \eta_i(j)\, N_j(0,1) \tag{2.1}$$

$$\eta_i'(j) = \eta_i(j) \exp(\tau' N(0,1) + \tau N_j(0,1)) \tag{2.2}$$

Where $x_i(j)$, $x_i'(j)$, $\eta_i(j)$ and $\eta_i'(j)$ denote the j-th component of the vectors x_i, x_i', η_i and η_i', respectively. N(0,1) denote a normally distributed one-dimensional random number with mean 0 and standard deviation 1. $N_j(0,1)$ indicates that a new random number is generated for each value of j. The τ and τ' are commonly set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$ [7, 8].

4. Calculate the fitness of each offspring $\left(x_i', \eta_i'\right), \forall i \in \{1,2,...,\mu\}$.
5. Conduct pair wise comparison over the union of parents (x_i, η_i) and offspring $\left(x_i', \eta_i'\right), \forall i \in \{1,2,...,\mu\}$. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is no smaller than the opponent's, it receives a "win."
6. Select the μ individuals, out of (x_i, η_i) and $\left(x_i', \eta_i'\right), \forall i \in \{1,2,...,\mu\}$, that have the most wins to be parents of the next generation.
7. Stop if the halting criterion is satisfied; otherwise, k=k+1 and go to Step 3.

2.2 Fast Evolutionary Programming (FEP)

The one-dimensional Cauchy density function centered at the origin is defined by:

$$f_t(x) = \frac{1}{\pi} \frac{t}{t^2 + x}, \qquad -\infty < x < \infty \tag{2.3}$$

where t>0 is a scale parameter. The corresponding distribution function is

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right).$$

The FEP studied in this paper is exactly the same as CEP described in last section except for Eq. (2.1) which is replace by the following [6]:

$$x_i'(j) = x_i(j) + \eta_i(j)\ \sigma_j \tag{2.4}$$

Where σ_j is a Cauchy random variable with the scale parameter t=1, and is generated anew for each value of j. It is worth indicating that we leave Eq. (2.2) unchanged in FEP in order to keep our modification of CEP to a minimum. It is also easy to

investigate the impact of the Cauchy mutation on EP when other parameters are kept the same.

2.3 Improved Fast Evolutionary Programming

Generally, Cauchy mutation performs better when the current search point is away from the global minimum, while Gaussian mutation is better at finding a local optimum in a given region. It would be ideal if Cauchy mutation is used when search points are far away from the global optimum and Gaussian mutation is adopted when search points are in the neighborhood of the global optimum. Unfortunately, the global optimum is usually unknown in practice, making the ideal switch from Cauchy to Gaussian mutation very difficult. Self-adaptive Gaussian mutation [7, 2, 9] is an excellent technique to partially address the problem. That is, the evolutionary algorithm itself will learn when to "switch" from one step size to another. However, there is room for further improvement to self-adaptive algorithms like CEP or even FEP.

In [19] an improved FEP (IFEP) based on *mixing* (rather than switching) different mutation operators. The idea is to mix different search biases of Cauchy and Gaussian mutation. The implementation of IFEP is very simple. It differs from FEP and CEP only in Step 3 of the algorithm described in Section 2.1. Instead of using Eq. (2.1) (for CEP) or Eq. (2.4) (for FEP) alone, IFEP generates two offspring from each parent, one by Cauchy mutation and the other by Gaussian. The better one is then chosen as the offspring. The rest of the algorithm is exactly the same as FEP and CEP. Chellapilla [10] has presented some more results on comparing different mutation operators in EP.

3 Optimum Search Direction Evolutionary Programming (OSDEP)

3.1 PSO Algorithms

Particle swarm optimization (PSO) algorithm was proposed by Kennedy and Eberhart [20] in 1995, which can be used to solve a wide array of different optimization problem. The PSO idea is inspired by natural concepts such as fish schooling, bird flocking and human social relations. PSO algorithm uses a number of particles that constitute a swarm. Each particle traverses the search space looking for the global optimum. In PSO system, particles fly in a multidimensional search space. During flight, each particle adjusts its position according to its own experience and the experience of neighboring particles, making use of the best position encountered by itself and its neighbors.

Assuming that the search space is D-dimensional, the swarm's size is N, the i-th particle of the swarm is represented by the D-dimensional vector $x_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ and the best particle of the swarm is denoted by index g. The best previous position of the i-th particle is recorded and represented by $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and the velocity of the i-th particle is $v_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot r_{i1}(t) \cdot [p_{id}(t) - x_{id}(t)] + c_2 \cdot r_{i2}(t) \cdot [p_{gd}(t) - x_{id}(t)] \quad (3.1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3.2)$$

Where, $d = 1, 2, \cdots, D$, t is the iteration number; $i = 1, 2, \cdots, N$; w is the inertia weight; c_1 and c_2 are two positive constants, called the cognitive and social parameter respectively; $r_{i1}(t)$ and $r_{i2}(t)$ are two random numbers uniformly distributed within the range [0,1]. The termination criterion for the iterations is determined according to whether the max generation or a designated value of the fitness of p_g is reached.

3.2 EP Based on Optimum Search Direction

Exploration ability is related to the algorithm's tendency to explore new regions of the search space; in this stage we should increase the search step size according to the search direction. Exploitation is the tendency to search a smaller region more thoroughly, in this stage; we should reduce the search step size. Researchers in PSO community have analyzed it empirically [22] and theoretically [21], they have shown the particles oscillate in different sinusoidal waves and converging quickly, sometimes prematurely, especially for PSO with small. w . They used to believe that inertia weight balances exploration and exploitation in PSO algorithm. In our opinion inertia weight can not balance exploration and exploitation. The factor to balance exploration and exploitation should be the velocity of particle. According to the global optimization search strategy, search direction and search step size are two important factors to affect the performance of algorithms. During PSO search, the current global best position gbest and the current best position of particles pbest indicate the search direction. In this paper we focus on analyzing how to control the search direction of EP, we introduce the global best position gbest into EP to control the search direction along with the current global optimum. The main steps of the OSDEP algorithm are as follows:

$$x_i'(j) = x_i(j) + \eta_i(j)\sigma_j + \eta_i(j)\mathrm{N}_j(0,1) \cdot (gbest - x_i(j)) \quad (3.3)$$

where $\mathrm{N}_j(0,1)$ indicates that a new random number is generated for each value of j, σ_j is a Cauchy random variable with the scale parameter t=1, and is generated anew for each value of j. gbest is the current global optimum.

It is also worth indicating that we leave Eq.(2.2) unchanged in OSDEP in order to keep our modification of CEP to a minimum. It is also easy to investigate the impact of the Cauchy mutation with optimum search direction on EP when other parameters are kept the same.

4 Benchmark Functions

The availability of appropriate and standardized sets of benchmark functions is very important for assessing evolutionary algorithms with respect to their effectiveness and efficiency [11].

Seven benchmark functions, from different sources, [2, 7, 12, 13] were used in our experimental studies. We assume that number of test problem considered in many other empirical studies carried out the past. However, this is necessary since the aim of this research is not to show OSDEP is better or worse IFEP, FEP and CEP, but to find when OSDEP is better (or worse) than IFEP, FEP, CEP and why. Wolpert and Macready [14] have shown that under certain assumptions no single search algorithm is better on average for all problems. If the number of test problems is small, it would be very difficult to generalize the claim. Using too small a test set also has the potential risk that the algorithm is biased (optimized) towards the chosen problems, while such bias might not be useful for other problem of interest.

Table 1. The 7 benchmark functions used in our experimental study, where n is the dimension of the function; f_{min} is the minimum value of the function, and $S \subseteq R^n$

Test Function	n	S	f_{min}
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	0
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	$[-10,10]^n$	0
$f_3(x)=$ -20exp $\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ - $\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)$ +20+e	30	$[-32,32]^n$	0
$f_4(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	$[-600,600]^n$	0
$f_5(x)=-\sum_{i=1}^{5}[(x-a_i)(x-a_i)^T+c_i]^{-1}$	4	$[0,10]^n$	-10
$f_6(x)=-\sum_{i=1}^{7}[(x-a_i)(x-a_i)^T+c_i]^{-1}$	4	$[0,10]^n$	-10
$f_7(x)=-\sum_{i=1}^{10}[(x-a_i)(x-a_i)^T+c_i]^{-1}$	4	$[0,10]^n$	-10

We summarize the benchmark functions which have been used to investigate the behavior of evolutionary algorithms in the continuous parameter optimization domain. The benchmark problems contain functions from simple unimodal to multimodal with few or many local optima. They range from low to high dimensional and scalable problems. A regular arrangement of local optima, separability of the objective function, decreasing difficultly of the problem with increasing dimensionality, and potential bias introduced by locating the global optimum at the origin of the coordinate system are identified as properties of multi-modal objective functions which are neither representative of arbitrary problems nor well suited for assessing the

global optimization qualities of evolutionary algorithms. The 7 benchmark functions are given in Table 1.

Functions f1 and f2 are high-dimensional problems, which are unimodal. Function f3 and f4 are multi-modal functions where the number of local minima increases exponentially with the problem dimension [12, 13]. Those classes of functions appear to be the most difficult class of problems for many optimization algorithms (including EP). Functions f5 to f7 are low-dimensional functions which have only a few local minima [13].

5 Experimental Studies and Discussions

5.1 Experimental Studies

In order to carry out a fair comparison among IFEP, FEP and CEP, the population size of IFEP was reduced to half of the FEP or CEP in all the following experiments, since each individual in IFEP generates two offspring. However, reducing IFEP's population size by half actually disadvantages IFEP slightly because it does no double the time for any operator (such as selection) other than mutations. Nevertheless, such comparison offers a good and simple compromise.

Table 2. Comparison among OSDEP, IFEP, FEP and CEP on function f1 to f7

F	#.of Gen.	OSDEP Mean Best	IFEP Mean Best	FEP Mean Best	CEP Mean Best
f1	1500	3.69×10^{-6}	4.16×10^{-5}	5.72×10^{-4}	1.91×10^{-4}
f2	2000	2.14×10^{-3}	2.44×10^{-2}	7.60×10^{-2}	2.29×10^{-2}
f3	1500	8.38×10^{-4}	4.83×10^{-3}	1.76×10^{-2}	8.79
f4	2000	9.86×10^{-3}	4.54×10^{-2}	2.49×10^{-2}	8.13×10^{-2}
f5	100	-10.15	-6.46	-5.50	-6.43
f6	100	-10.40	-7.10	-5.73	-7.62
f7	100	-10.54	-7.8	-6.41	-8.86

IFEP was tested in the same experimental setup as FEP and CEP. In all experiments, the same self-adaptive method (i.e., Eq.(2.2)), the same population size, μ=100, the same tournament size, q=10, for selection, the same initial η =3.0, and the same initial population were used for OSDEP, IFEP, FEP and CEP. These parameters were considered as suggested by Bäck and Schwefel[7] and Fogel[2]. The average results of 50 independent runs are summarized in Table 2.

All results have been average over 50 runs, where "Mean Best" indicates the mean best function values found in the last generation.

5.2 Discussions

It is very clear from Table 2 that OSDEP has improved EP's performance significantly for all test functions. These results show that OSDEP continues to perform

much better than IFEP on multi-modal functions with many minima, and also performs very well on unimodal functions and multi-modal functions with only a few local minima, which IFEP and FEP have difficulty in handling. OSDEP achieved performance also much better than CEP's on these functions.

For the two unimodal functions where FEP is outperformed by CEP significantly, OSDEP performs better than CEP on f1 and f2. A close look at the actual average solutions results that OSDEP found much better solution than CEP on f1 and f2 roughly an order of magnitude smaller.

For the three Shekel function f5 to f7, the difference between IFEP and CEP is much smaller than that between FEP and CEP. IFEP has improved FEP's performance significantly on all three functions. It performs better than CEP on f5 and f6 and worse on f7. OSDEP's performance is the best for all the EP versions.

It is very encouraging that OSDEP is capable of performing as well as or better than the better one of IFEP, FEP and CEP for all the test functions. This is achieved through a minimal change to the existing FEP and CEP. No prior knowledge or any complicated operators were use, and also no additional parameter was used either. The superiority of OSDEP also demonstrates the importance of mixing different search biases (e.g. "step sizes") and optimum search direction in a robust search algorithm.

It is that Cauchy mutations played a major role in the population in the early stages of evolution since the distance between the current search points and the global optimum was relatively large on average in the early stages, hence Cauchy mutation performed better. However, as the evolution progress, the distance became smaller and smaller. Large search step sizes produced by Cauchy mutation tended to produce worse offspring than those produced by Gaussian mutation.

6 Conclusions and the Future Work

OSDEP uses the idea of mixing search biases to mix Cauchy and Gaussian mutations and the current global optimum search direction. Unlike some switching algorithms which have to decide when to switch between different mutations during search, OSDEP does not need to make such decision and introduces no parameters. OSDEP is robust, assumes no prior knowledge of the problem to solved, and performs much better than IFEP, FEP and CEP for all the benchmark problems. Future work on OSDEP includes the comparison of OSDEP with other self-adaptive algorithms such as [15] and other evolutionary algorithms [16].

The hybrid algorithm is one kind of heredity optimization technique based on the swarm intelligence; the whole population evolves not only through selection and mutation operators, but also fly directed by individual's memory and the desire to be better. So the new algorithm can rapid find the best value.

For hybrid algorithm, the new population has the very good individual because of migration mechanism. The good individual can accelerate the convergence rate, jump out the local optimum points using mix Cauchy and Gaussian mutation.

The idea of FEP and OSDEP can also be applied to other evolutionary algorithms to design faster optimization algorithms [17]. For $(\mu+\lambda)$ and (μ,λ) evolutionary algorithms where $\mu<\lambda$, IFEP would be particularly attractive since a parent has to generate more than one offspring. It may be beneficial if different offspring are generated by different mutations [17].

Acknowledgements

This paper supported by the National Natural Science Foundation of China (No. 60772163), (No.60473081) and Shenzhen Science and Technology Project (No. SZKJ0706).

References

1. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence Through Simulated Evolution. John Wiley & Sons, New York (1966)
2. Fogel, D.B.: System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling. Ginn Press, Needham Heights (1991)
3. Fogel, D.B.: Evolving Artifical Intelligence. PhD thesis, University of California, San Diego, CA (1992)
4. Fogel, D.B.: Applying evolutionary programming to selected traveling samlesman problems. Cybernetics and Systems 24, 27–36 (1993)
5. Yao, X.: An overview of evolutionary computation. Chinese Journal of Advanced Software Research 3(1), 12–29 (1996)
6. Yao, X., Liu, Y.: Fast Evolutionary Programming. In: Fogel, L.J., Angeline, P.J., Bäck, T. (eds.) Evolutionary Programming V: Proc. of the Fifth Annual Conference on Evolutionary Programming, pp. 257–266. The MIT Press, Cambridge (1996)
7. Bäck, T., Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation 1(1), 1–23 (1993)
8. Fogel, D.B.: An Introduction to Simulated Evolutionary Optimization. IEEE Trans. On Neural Networks 5(1), 3–4 (1994)
9. Fogel, D.B.: Evolutionary computation: Towards a new philosophy of machine intelligence. IEEE Press, New York (1995)
10. Chellapilla, K.: Combining mutation operators in evolutionary programming. IEEE Trans. On Evolutionary Computation 2(3), 91–96 (1996)
11. Bäck, T., Fogel, D.B., Michalewicz, Z.: Handbook of Evolutionary Computation. IOP Publishing/Oxford University Press (1997)
12. Schwefel, H.-P.: Evolution and Optimum Seeking. John Wiley & Sons, New York (1995)
13. Törn, A., Zilinskas, A.: Global Optimization. In: Törn, A., Žilinskas, A. (eds.) Global Optimization. LNCS, vol. 350. Springer, Heidelberg (1989)
14. Wolpert, D.H., Macready, W.G.: No free lunch theorems for search. IEEE Transcation on Evolutionary Computation 1(1), 67–82 (1997)
15. Omran, M.G.H., Salman, A., Engelbrecht, A.P.: Self-adaptive Differential Evolution. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS, vol. 3801, pp. 192–199. Springer, Heidelberg (2005)
16. Lam, T., Soliman, O., Abbass, H.A.: A Modified Strategy for the Constriction Factor in Particle Swarm Optimization. In: Randall, M., Abbass, H.A., Wiles, J. (eds.) ACAL 2007. LNCS, vol. 4828, pp. 333–344. Springer, Heidelberg (2007)
17. Yao, X., Liu, Y.: Fast Evolutiona Strategies. Control and Cybernetics 26(3), 467–496 (1997)
18. Duan, M., Povinelli, R.: Nonlinear Modeling: Genetic Programming vs. Fast Evolutionary Programing. In: Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE 2001), pp. 171–176 (2001)

19. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evolutionary Computation 3(2), 82–102 (1999)
20. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. IEEE International Conference on Neural Networks, IV, vol. 1, pp. 942–948. IEEE Service Center, Piscataway (1995)
21. Clerk, M., Kennedy, J.: The particle swarm explosion, stability and convergence in a multi-dimensional complex space. IEEE Trans. Evol. Comput. 6(1), 58–63 (2002)
22. Angeline, P.J.: Evolutionary optimization versus particle swarm optimization: philosophy and performance differences. Evolutionary programming 7, 601–610 (1998)
23. Lee, C.Y., Yao, X.: Evolutionary programming using the mutations based on the Levy probability distribution. IEEE Transactions on Evolutionary Computation 8(1), 1–13 (2004)

Dynamic Clonal and Chaos-Mutation Evolutionary Algorithm for Function Optimization

Ming Yang[1] and Jing Guan[2]

[1] School of Computer Science, China University of Geosciences, Wuhan, 430074, China
yangming0702@gmail.com
[2] Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, 430074, China
g_jing0414@yahoo.com.cn

Abstract. This paper introduced a dynamic-clone and chaos-mutation evolutionary algorithm (DCCM-EA), which employs dynamic clone and chaos mutation methods, for function optimization. The number of clone is direct proportion to "affinity" between individuals and the chaos sequence can search the points all over the solution space, so DCCM-EA can make all points get equal evolutionary probability, to get the global optimal solution most possibly. In the experiments, taking 23 benchmark functions to test, it can be seen that DCCM-EA if effective for solving function optimization.

Keywords: function optimization**,** evolutionary algorithm, dynamic clone, chaos mutation.

1 Introduction

In real-life world, there are lots of function optimization problems, especially with multi-peaks, so the works on this field can make great sense. The content of function optimization is to search the best point, which is the minimum point commonly, from all the search space. For these problems to get the maximum point, they can make minus and are changed to search the minimum point. The minimum function optimization problem can be described as follows:

$$\min f(x), x \in D \tag{1}$$

where f is an n-dimension function, D is a limited space and $D \subseteq R^n$.

A number of evolutionary algorithms have been used to solve these function optimization problems [1-6]. This success is due to EAs essentially are search algorithms based on the concepts of natural selection and survival of the fittest. They guide the evolution of a set of randomly selected individuals through a number of generations in approaching the global optimum solution. Besides that, the fact that these algorithms do not require previous considerations regarding the problem to be optimized and offers a high degree of parallelism is also true. Especially, Evolutionary Programming (EP) has been applied with success to many function optimization problems in recent years [3]. Xin Yao has proposed a "fast EP" (FEP) which uses a Cauchy instead of

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 19–27, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Gaussian mutation as the primary search [4]. STGA, which is based on a robust stochastic strategy, can get very good results for many function optimization problems [5]. Quantum-inspired evolutionary algorithm, supplemented by quantum method, can also solve many function optimization problems effectively [6].

Some new algorithms such as Particle Swarm Optimization (PSO) [7-9] and Differential Evolution (DE) [10-12] are proposed to solve function optimization problems. PSO is an extremely simple algorithm that seems to be effective for optimizing a wide range of functions. Unique to the concept of particle swarm optimization is flying potential solutions through hyperspace, accelerating toward "better" solutions. Much of the success of particle swarms seems to lie in the agents' tendency to hurtle past their target. DE is a practical approach to global function optimization that is easy to understand, simple to implement, reliable, and fast. Basically, DE adds the weighted difference between two population vectors to a third vector. This way no separate probability distribution has to be used which makes the scheme completely self-organizing.

In this paper, a dynamic-clone and chaos-mutation evolutionary algorithm (DCCM-EA) for function optimization is proposed, using dynamic clone and chaos mutation methods. The number of clone is direct proportion to "affinity" between individuals and the chaos sequence can search the points all over the solution space, so DCCM-EA can make all points get equal evolutionary probability, to get the global optimal solution.

2 Dynamic-Clone and Chaos-Mutation Evolutionary Algorithm (DCCM-EA)

Suppose that $\boldsymbol{A}(k) = \{A_1(k+1), A_2(k+1), ..., A_n(k+1)\}$ is the population at the k generation and n is the size of population. The individual $\boldsymbol{A}(k)$ is coded by real number:

$$A_i(k) = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, 2, ..., n \tag{2}$$

where m is the dimension of function and $x_{ij} \in [a_j, b_j]$, a_j, b_j is the boundary of x_j.

DCCM-EA clones an individual according to "affinity" between individuals, operates chaos mutation on the population and then generation the offspring population. The course of DCCM-EA is as follow:

$$\boldsymbol{A}(k) \xrightarrow{T_c} \boldsymbol{Y}(k) \xrightarrow{T_g} \boldsymbol{A}(k) \cup \boldsymbol{Y'}(k) \xrightarrow{T_s} \boldsymbol{A}(k+1) \tag{3}$$

where T_c is the dynamic clone operation, T_g is the evolutionary operation and T_s is the selection operation to generate offspring population. The three operations are described as Fig.1.

2.1 Dynamic Clone Operation T_c

Suppose that $\boldsymbol{Y}(k)$ is the clone population after T_c. $\boldsymbol{Y}(k)$ is:

$$\boldsymbol{Y}(k) = T_c(\boldsymbol{A}(k)) = \left[Y_1(k) \quad Y_2(k) \quad \cdots \quad Y_n(k) \right]^{\mathrm{T}} \tag{4}$$

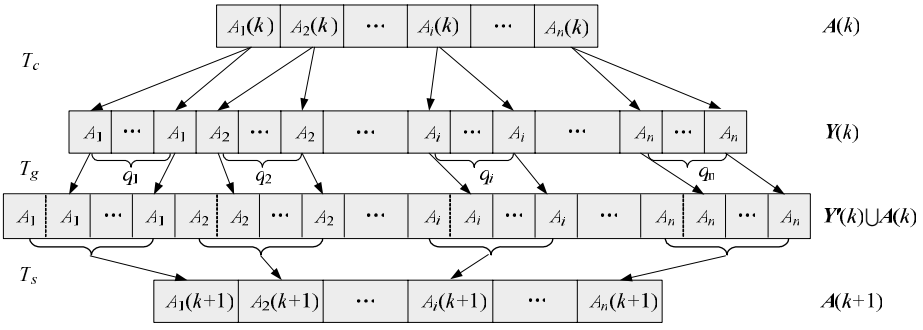


Fig. 1. The three operations of DCCM-EA: T_c is the dynamic clone operation, T_g is the genetic operation and T_s is the operation to generate offspring population

where $Y_i(k) = T_c(A_i(k)) = \boldsymbol{I}_i \times A_i(k), i = 1,2,...,n$, $\boldsymbol{I}_i$ is the q_i-dimension row vector in which each element is 1 and the value of q_i is:

$$q_i = \lceil low + (up - low) \times \Theta_i \rceil \tag{5}$$

where *low* and *up* is respectively the minimum and maximum value of clone size, Θ_i is the affinity between A_i and other individuals and $\Theta_i \in [0,1]$:

$$\Theta_i = \frac{d_i - d_{\min}}{d_{\max} - d_{\min}} \tag{6}$$

where d_i is the Euclidian distance between A_i and other individuals A_j:

$$\begin{aligned} d_i &= \min \|A_i - A_j\|, j = 1,2,...,n, i \neq j \\ d_{\min} &= \min\{d_i\}, i = 1,2,...,n \\ d_{\max} &= \max\{d_i\}, i = 1,2,...,n \end{aligned} \tag{7}$$

q_i changes with Θ_i which indicates how dense the population's distribution is near the individual A_i. The more dense distribution, the less Θ_i is, vice versa.

Suppose that $\boldsymbol{Y}(k) = \{Y_1(k), Y_2(k), ..., Y_n(k)\}$ is the clone population, where $Y_i(k) = [A_i(k) \quad A_i(k) \quad \cdots \quad A_i(k)]_{1 \times q_i}$.

Suppose that $\boldsymbol{Y}'(k) = \{Y_1'(k), Y_2'(k), ..., Y_n'(k)\}$ is the population after the clone operation T_c.

According to affinity between individuals, T_c can make the more q_i in the domain where the population's distribution is dense and the less q_i in the domain where the population's distribution is loose. So DCCM-EA can make all points in the search space get equal evolutionary probability, to get the global optimal solution.

2.2 Chaos Mutation Operation T_g

2.2.1 Chaos

Chaos is seemingly a kind of irregular motions, which has random-like behavior without any random factors in the deterministic non-linear system (randomness of the inner algorithm framework). The greatest characteristic of chaotic system is that it is very sensitive to the initial value, and the future behavior of system is unpredictable in long term. We can apply these characteristics of chaos to the optimization calculation, chaotic regularity makes its definitive iteration formula generate the new solutions easily, and the searching can avoid to fall into the local optimum by chaotic randomness, the most important is that the final solution can approach to the true optimal solution with arbitrary precision by chaotic ergodicity. Random number generated by ordinary methods can't complete the ergodic search over continuous variable spaces because it can't reach arbitrary precision without repetition, but chaotic variables are feasible during the precision range that computer can represent.

In this paper, the chaos function is Logistic function [13], which is a classical chaos system:

$$r_{n+1} = \mu r_n \left(1 - r_n\right), n = 0,1,2,... \tag{8}$$

where μ is a parameter. It can generate a serial sequence when the first number $r_0 \in [0,1]$ is randomly produced and μ is specific.

The Logistic appears different characteristic with Different value of μ (see Fig.2). When μ=4, Logistic system is in the chaos state and $r_n \in [0,1]$ is chaos sequence which can search the points all over the solution space. Because of the characters of Logistic system, it can be applied in the optimization field.

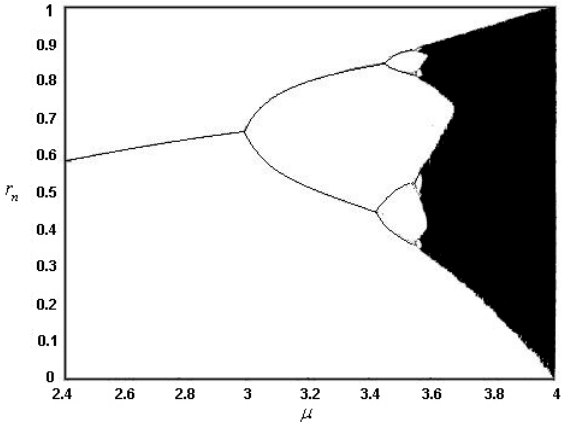


Fig. 2. How Logistic enters chaos. The Logistic appears different characteristic with Different value of μ.

2.2.2 Chaos Mutation Operator T_g

The new individual is $A_i'(k)=(x_{i1}',x_{i2}',...,x_{im}')$ after the chaos mutation operation T_g, $\forall A_i(k)\in Y_i(k)$, $A_i(k)=(x_{i1},x_{i2},...,x_{im})$. The following is the details of T_g.

```
for (1 ≤ j ≤ m)
{
  if (random(0,1)<P_m)
   {
     if (random(0,1)<0.5)
        x'_ij = x_ij + (r_ij × (b_ij − a_j) + a_j), 1 ≤ j ≤ m;
     else
        x'_ij = x_ij − (r_ij × (b_ij − a_j) + a_j), 1 ≤ j ≤ m;
     if (x'_ij < a_j)  x'_ij = a_j;
     else if (x'_ij > b_j)  x'_ij = b_j;
   }
}
```

where r_{ij} is the chaos sequence of Logistic system when μ=4; $P_m=m^{-(n+1)/(n+t_i)}$, $P_m\in\left[\frac{1}{m},\sqrt{\frac{1}{m}}\right]$; t is the ranking of $A_i(k+1)$ in the sort sequence according to fitness.

Relative to the individual of better fitness, the ones of worse fitness can get greater probability. This can guarantee the whole population is good.

2.3 Selection Operation T_s

$\forall i=1,2,...,n$, denote $B_i(k)=best\{Y_i'(k)\}=best\{A_i'(k)\mid i=1,2,...,q_i\}$. After chaos mutation operation, the offspring population is:

$$A_i(k+1)=\begin{cases}B_i(k) & \text{if } B_i(k) \text{ is better than } A_i(k)\\ A_i(k) & \text{else}\end{cases} \tag{9}$$

2.4 The Framework of DCCM-EA

1. Suppose that k is the number of evolution, k=0;
2. Initialize the population $\boldsymbol{A}(k)$ randomly, the size of population is n;
3. Initialize the first number of Logistic chaos sequence r_0;
4. Dynamic clone operation T_c for each individual in $\boldsymbol{A}(k)$, generate $\boldsymbol{Y}(k)$;
5. Chaos mutation operation T_g for each individual in $\boldsymbol{Y}(k)$, generate $\boldsymbol{Y}'(k)$;
6. Selection Operation T_s, generate the offspring population $\boldsymbol{A}(k+1)$;

7. k:=k+1;
8. If not termination condition, go to step 4;
9. Stop.

3 Experiments

The current version of DCCM-EA is proposed to solve real-parameter problems. In this paper, it is used to solve the complex numerical problems. The benchmark functions here are a set of 23 different functions selected from [4]. These functions are widely used as benchmark in numerical optimization. Functions f_1-f_{13} are high dimensional problems. Functions f_1-f_5 are uni-modal. Function f_6 is the step function, which has one minimum and is discontinuous. Function f_7 is a noisy quartic function, where the random number is a uniformly distributed random variable in [0, 1). Functions f_8-f_{13} are multi-modal functions where the number of local minima increases exponentially with the problem dimension [14,15]. Functions f_{14}-f_{23} are low-dimensional

Table 1. The 23 benchmark functions used in experiment, where N is the dimension of function, f_{min} is the minimum value of function, and S is the search space, $x \in S$

Function	N	S	f_{min}
f_1	30	$[-100, 100]^N$	0
f_2	30	$[-10, 10]^N$	0
f_3	30	$[-100, 100]^N$	0
f_4	30	$[-100, 100]^N$	0
f_5	30	$[-30, 30]^N$	0
f_6	30	$[-100, 100]^N$	0
f_7	30	$[-1.28, 1.28]^N$	0
f_8	30	$[-500, 500]^N$	-12569.5
f_9	30	$[-5.12, 5.12]^N$	0
f_{10}	30	$[-32, 32]^N$	0
f_{11}	30	$[-600, 600]^N$	0
f_{12}	30	$[-50, 50]^N$	0
f_{13}	30	$[-50, 50]^N$	0
f_{14}	2	$[-65.536, 65.536]^N$	1
f_{15}	4	$[-5, 5]^N$	0.0003075
f_{16}	2	$[-5, 5]^N$	-1.0316285
f_{17}	2	$[-5, 10] \times [0, 15]$	0.398
f_{18}	2	$[-2, 2]^N$	3
f_{19}	3	$[0, 1]^N$	-3.86
f_{20}	6	$[0, 1]^N$	-3.32
f_{21}	4	$[0, 10]^N$	-10
f_{22}	4	$[0, 10]^N$	-10
$\boldsymbol{f_{23}}$	4	$[0, 10]^N$	-10

functions which have only a few local minima [14]. In order to compare DCCM-EA with FEP [4], the setting of the 23 functions, which are given in Table 1, is the same as in [4]. The more details of each function can be got from [4].

The parameters of algorithm are set as follow: n=50, low=10 and up=20. The results of performing DCCM-EA are listed in Table 2. Table 2 shows best, worst and mean best values for each function evaluated for independent 50 runs.

In this paper, DCCM-EA is compared with FEP which uses a Cauchy instead of Gaussian mutation as the primary search operator. DCCM-EA uses chaos mutation and the mutation probability is a self-adaptive value. The mutation operation is feasible for different functions. The results of this comparison are shown in Table 3 with respect to the mean best values found and the standard deviations for each function between DCCM-EA and FEP. From the results, it can be seen that the results of DCCM-EA is not better than FEP for f_3 and is little worse than FEP for f_{15}, but it is

Table 2. The best, worst and mean best values for each function got by DCCM-EA. Each functoin is evaluated independently for 50 runs.

Function	*Number of Generations*	*Best*	*Worst*	*Mean Best*
f_1	50000	1.216743×10^{-6}	5.870576×10^{-6}	2.980852×10^{-6}
f_2	50000	4.174004×10^{-4}	7.921635×10^{-4}	5.899849×10^{-4}
f_3	50000	15.34696	35.14507	25.31694
f_4	50000	1.319475×10^{-1}	1.962558×10^{-1}	1.627719×10^{-1}
f_5	50000	2.920885×10^{-2}	2.432035	4.151956×10^{-1}
f_6	20000	0	0	0
f_7	50000	2.601439×10^{-3}	6.193726×10^{-3}	4.960406×10^{-3}
f_8	20000	-12569.49	-12569.49	-12569.49
f_9	50000	5.329253×10^{-7}	2.691468×10^{-6}	1.678072×10^{-6}
f_{10}	50000	2.621593×10^{-4}	6.431235×10^{-4}	4.108969×10^{-4}
f_{11}	50000	2.025077×10^{-6}	2.411329×10^{-5}	8.193575×10^{-6}
f_{12}	50000	3.418652×10^{-9}	1.248548×10^{-8}	8.268964×10^{-9}
f_{13}	50000	4.454944×10^{-8}	2.429853×10^{-7}	1.145081×10^{-7}
f_{14}	100	0.9980038	0.9980038	0.9980038
f_{15}	50000	3.091611×10^{-4}	6.889355×10^{-4}	5.493655×10^{-4}
f_{16}	200	-1.031628	-1.031628	-1.031628
f_{17}	500	0.3978874	0.3978876	0.3978875
f_{18}	500	3	3	3
f_{19}	200	-3.862782	-3.862782	-3.862782
f_{20}	200	-3.321972	-3.321661	-3.321848
f_{21}	1000	-10.15320	-10.15318	-10.15319
f_{22}	1000	-10.40294	-10.40290	-10.40293
f_{23}	1000	-10.53641	-10.53639	-10.53640

Table 3. Comparison between DCCM-EA and FEP, where "Mean Best" and "Std Dev" stand for the mean best values and the standard deviation respectively. Each function is evaluated independently for 50 runs.

Function	*DCCM-EA*		*FEP*	
	Mean Best	Std Dev	Mean Best	Std Dev
f_1	2.980852×10^{-6}	1.159304×10^{-6}	5.7×10^{-4}	$1{,}3\times10^{-4}$
f_2	5.899849×10^{-4}	1.006117×10^{-4}	8.1×10^{-3}	7.7×10^{-4}
f_3	25.31694	6.624966	1.6×10^{-2}	1.4×10^{-2}
f_4	1.627719×10^{-1}	2.049367×10^{-2}	0.3	0.5
f_5	4.151956×10^{-1}	6.832193×10^{-1}	5.06	5.87
f_6	0	0	0	0
f_7	4.960406×10^{-3}	1.088628×10^{-3}	7.6×10^{-3}	2.6×10^{-3}
f_8	-12569.49	0	-12554.5	52.6
f_9	1.678072×10^{-6}	6.011691×10^{-7}	4.6×10^{-2}	1.2×10^{-2}
f_{10}	4.108969×10^{-4}	1.029389×10^{-4}	1.8×10^{-2}	2.1×10^{-3}
f_{11}	8.193575×10^{-6}	5.940091×10^{-6}	1.6×10^{-2}	2.2×10^{-2}
f_{12}	8.268964×10^{-9}	2.655438×10^{-9}	9.2×10^{-6}	3.6×10^{-6}
f_{13}	1.145081×10^{-7}	5.942774×10^{-8}	1.6×10^{-4}	7.3×10^{-5}
f_{14}	0.9980038	0	1.22	0.56
f_{15}	5.493655×10^{-4}	1.119569×10^{-4}	5.0×10^{-4}	3.2×10^{-4}
f_{16}	-1.031628	0	-1.03	4.9×10^{-7}
f_{17}	0.3978875	4.488186×10^{-8}	0.398	1.5×10^{-7}
f_{18}	3	0	3.02	0.11
f_{19}	-3.862782	0	-3.86	1.4×10^{-5}
f_{20}	-3.321848	1.003577×10^{-4}	-3.27	5.9×10^{-2}
f_{21}	-10.15319	9.157152×10^{-6}	-5.52	1.59
f_{22}	-10.40293	1.176612×10^{-5}	-5.52	2.12
f_{23}	-10.53640	5.140836×10^{-6}	-6.57	3.14

apparent that DCCM-EA performs better than FEP for the rest part of the 23 benchmark functions.

4 Conclusions

This paper introduced a dynamic-clone and chaos-mutation evolutionary algorithm (DCCM-EA), which employs dynamic clone and chaos mutation methods, for function optimization. According to affinity between individuals, clone operation can make the more clones in the domain where the population's distribution is dense and the less clones in the domain where the population's distribution is loose. So DCCM-EA can make all points in the search space get equal evolutionary probability, to get the global optimal solution. The chaos mutation implying sequence generated by

Logistic chaos system can search the points all over the solution space, which can avoid falling into the local optimum.

References

1. Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Institute to Physics Publishing (1997)
2. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs, 2nd edn. Springer, New York (1994)
3. Gehlhaar, D.K., Fogel, D.B.: Tuning evolutionary programming for conformationally flexible molecular docking. In: Fogel, L.J., Angeline, P.J., Bäck, T. (eds.) Evolutionary Programming V: Proc. Of the Fifth Annual Conference on Evolutionary Programming, pp. 419–429. MIT Press, Cambridge (1996)
4. Yao, X., Liu, Y., Lin, G.M.: Evolutionary programming made faster. IEEE Trans. on Evolutionary Computation 3, 82–102 (1999)
5. Tu, Z., Lu, Y.: A robust stochastic genetic algorithm (stga) for global numerical optimization. IEEE Trans. on Evolutionary Computation 8(5), 456–470 (2004)
6. André, V., da Curz, A., Vellasco, M.M.B.R.: Quantum-Inspired Evolutionary Algorithm for Numerical Optimization. In: 2006 IEEE Congress on Evolutionary Computation, pp. 9181–9187 (2006)
7. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. on Evolutionary Computation 6(1), 58–73 (2002)
8. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through particle swarm optimization. Natural Computing, 235–306 (2002)
9. Andrews, P.S.: An Investigation into Mutation Operators for Particle Swarm Optimization. In: IEEE Congress on Evolutionary Computation, pp. 3789–3796 (2006)
10. Storn, R., Price, K.V.: Differential evolution-A simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
11. Lampinen, J., Zelinka, I.: Mixed Variable Non-linear Optimization by Differential Evolution. In: Proceedings of Nostradamus 1999 2nd International Prediction Conference, pp. 45–55 (1999)
12. Price, K., Storm, R., Lampinen, J.: Differential Evolution-A Practical Approach to Global Optimization. Springer, Heidelberg (2005)
13. Heinx, O.P.: Chaos and Fractals. Springer, New York (1992)
14. Törn, A., Žilinskas, A.: Global Optimization. LNCS, vol. 350. Springer, Heidelberg (1989)
15. Schwefel, H.P.: Evolution and Optimum Seeking. Wiley, New York (1995)

A Parallel Self-adaptive Subspace Searching Algorithm for Solving Dynamic Function Optimization Problems

Yan Li[1], Zhuo Kang[1], and Lishan Kang[2]

[1] Computation Center, Wuhan University, Wuhan 430072, China
[2] School of Computer Science, China University of Geosciences(Wuhan)
Wuhan, 430074, China
kang_whu@yahoo.com

Abstract. In this paper, a parallel self-adaptive subspace searching algorithm is proposed for solving dynamic function optimization problems. The new algorithm called DSSSEA uses a re-initialization strategy for gathering global information of the landscape as the change of fitness is detected, and a parallel subspace searching strategy for maintaining the diversity and speeding up the convergence in order to find the optimal solution before it changes. Experimental results show that DSSSEA can be used to track the moving optimal solutions of dynamic function optimization problems efficiently.

Keywords: evolutionary algorithm, dynamic function optimization problem, niche strategy.

1 Introdction

The dynamic function optimization problem always handles changing function, that is to say that at least one part of the problem always changes with time, such as the optima, the fitness function, the searching space or the location of the optima. The optimization of solutions in dynamic environments is of particular importance in real-world problems[1]. Commonly, the dynamic function optimization problem can be described as follows

$$\text{Maximize } Y = f(X, t) \tag{1}$$

s.t.

$$g_i(X, t) \leqq 0,\ i = 1, 2, \ldots, k$$
$$h_j(X, t) = 0,\ j = 1, 2, \ldots, l$$
$$X \in D = \{ X \in R^n \mid X_l \leqq X \leqq X_u \}$$
$$t \in T = \{ t \in R \mid 0 \leqq t \leqq t^* \}$$
$$Y \in R^p \times T$$

where X is decision variable, Y is objective variable. When $p \geqq 2$, the expression (1) is called dynamic multi-objective function optimization problem; when $p = 1$ the expression (1) is called dynamic function optimization problem. And when $k+l > 0$, the expression (1) is called constrained dynamic function optimization problem; when

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 28–36, 2008.
© Springer-Verlag Berlin Heidelberg 2008

$k=l=0$, the expression (1) is called non-constrained dynamic function optimization problem. When $t^*=0$, the expression (1) is called multi-objective function optimization problem. In this paper, we only discuss the situation when $p=1$ and $k=l=0$.

2 Algorithm DSSSEA

The essence to solve fucntion optimization problem in dynamic environments is to detect the change of condition in the progress of seraching and to make the algorithm to catch up with the change of condition. It has been said that to find an effective way to solve dynamic function optimization problem, that is to say, improving diversity of individuals can make the algorithm to adapt to the change of condition more quickly. So, how to improve diversity of solutions? There are some methods which have been proven effective: reinitialization or partly reinitialization; hypermutation; to reserve good indivisuals from history; multi-population strategy and self-adaptive strategy and so on[2,3,4,5,6,7].

The creature has the ability to produce all kinds of new creatures because in the progress of evolution, the creature can divide into many small population (we call it niche), and the difference among these niches become bigger and bigger and then become into different creatures. Why the niche can cause division of population? There are three causes: firstly, the genes of the niches are not the same, especially for small population, at the beginning of the evolution the difference has existed in; secondly, the separated niches has utterly different mutation, some changes occur in this niche, other changes occur in other niches, the change of one niche doesn't influence other niches; thirdly, different choice, two different niches have different condition, different condition means different choice, which causes the niches develop in different ways.

We denote the feasible solution of problem (1) at time T by $X(T)$, the number of subspaces or niches by n.

The subspace and niche are defined as follows:

Definition 1. subspace $S_i^t(T)=\{X\in D \mid \|X-X_i^t(T)\|\le ih\}, i=1,2,\cdots,n$;

where h is a parameter used to control the size of subspace.

Definition 2. niche $n_i=\{X_{ij}^t(T)\}_{j=1}^m$.

where m is the niche size.

To solve dynamic function optimization problems, we use evolutionary algorithm with niche strategy which producing a set of niches by mutation, that means Dynamic Searching in Sub-Spaces, so we call this new algorithm DSSSEA. In DSSSEA, firstly the individuals are ordered according to their fitness values, then around the individuals (in subspaces) producing niches by mutation, but the size of subspace is inversely proportional to the fitness of individual. The strategy is to speed up the convergence and to maintain the diversity to avoid local optima and premature. This algorithm only uses objective fitness values without the limit of continuous function, so it can be used in many kinds of problems.

A multi-processor system with shared memory can be used to perform an asynchronous parallel algorithm efficiently. Suppose that the population $p^t(T)$ used as a global variable is stored in the shared memory . The algorithm DSSSEA is described as follows.

Algorithm DSSSEA

```
Begin
Step 1: T:=0;
Step 2: initialize P^0(T) = {X_1^0(T), X_2^0(T), ..., X_n^0(T)} randomly
from D;
Step 3: ranking the individuals of P^0(T) according to
their fitness;
Step 4: for i=1 to n do in parallel
          t:=0;
       L:  for j=1 to m d0
              X_ij^t(T) := X_i^t(T) + i*h*N(0,1)
           endfor
           X_i^{t+1}(T) := arg max_{0≤j≤m} f(X_ij^t(T));
           t := t+1;
           if  t<t_max  then goto L;
         endfor;
Step 5: output  X_best(T) = max_{0≤i≤n} f(X_i^{t_max}(T));
Step 6: T := T + ΔT;
Step 7: if T ≤ t* goto Step 2;
Step 8: stop.
End
```

In this program, the host processor is used to perform step 1,2,3,5,6 and 7, while others perform step 4 asynchronously. Besides, Step 4 is a parallel process used to maintain the diversity and to keep the capability of high convergence speed in order to find the optimal solution before the landscape changes. Step 5 is a synchronized point for output of results. Step 7 turns to reinitialize the population for learning the global information of landscape as the time changes in step 6 and to restart the dynamic subspace searching process.

Since the dynamic problem is much stricter with diversity of solutions than any other problems, if there is no any knowledge about the fitness scope, the *N* initialized individuals should be evenly located in the solution space in order to ensure the global searching and to improve the efficiency of algorithm.

Taking note of step 4, if we do it in sequence (not in parallel), then Algorithm DSSSEA becomes a sequential algorithm which can be performed with any computers.

3 Numerical Experiments

3.1 Test Function

The dynamic problems producer DF1[8] is used in our experiments. DF1 uses "two step" strategy: the first step is to produce a set of taper with different height and different slope (DF1 can be defined as any dimension); the second step is to add the

dynamic characteristics to these tapers. For instance, in two-dimension condition, the static function DF1 can be described as:

$$f(X,Y) = \max_{i=1,N}[H_i - R_i * \sqrt{(X - X_i)^2 + (Y - Y_i)^2}]$$

Where N is the number of tapers, and all of these tapers are independent for each other, they are all decided by their own location (X_i, Yi), height (H_i) and slope (R_I). The objective function $f(X, Y)$ is used to join all of the N tapers to form a complex function with N local maxima. The dynamic style of $f(X, Y)$ is controlled by the frequency of producing new taper by function producer. The function producer randomly produces N tapers in pre-decided range each time, such that:

$$H_i \in [Hbase, Hbase + Hrange];$$
$$R_i \in [Rbase, Rbase + Rrange];$$
$$X_i \in [-1,1];$$
$$Y_i \in [-1,1].$$

Where *Hbase* is the minimal height of the taper, *Hrange* is the range of taper height, *Rbase* is the minimal slope of the taper, *Rrange* is the range of taper slope.

Because programmers can decide the evolution generation in the test function DF1, we use the improved offline evaluation method *Acc*: to compute the average value of the fitness of the best solutions up to now, it can show how quickly the algorithm close to the optima. The mathematics expression is as follows:

$$Performance_offline = \frac{1}{K}\sum_{i=1}^{K}(f_i^*)$$

Where $f_i^* = \max\{f_1, f_2, ..., f_i\}$, that is to say f_i^* is the fitness of the best solution of all of the individuals up to the *i*th generation.

3.2 Choices of Parameters

The parameters we used in the test function DF1 are listed in the table 1:

Table 1. The parameters

parameters	5N2M5S	14N2M14C	5N5M5S	5N5M5C	14N10M14C
Hbase	60.0	1.0	60.0	60.0	1.0
Hrange	0.0	9.0	0.0	0.0	9.0
Rbase	70.0	8.0	70.0	70.0	8.0
Rrange	0.0	12.0	0.0	0.0	12.0
Acc	1.5	3.8	1.5	3.8	3.8

In the above table, for instance, 5N2M5S denotes that the number of taper is 5, the dimension N is 2, the number of dynamic apices M is 5, and S,C and L denote the change is small, chaos or large. Usually, we set the population size 100, the niche size

100, the function changing generation is 20, and the termination condition is that the generation equal to 400. In order to avoid the error caused by random algorithm, all of the solutions are the average value of ten runs.

3.3 The Numerical Experiments

In the following figures, the value of the vertical coordination is the performance evaluation *Acc*. We did all kinds of experiments on an PC.

1) Changing the size of population
For convenience, we set the population size as 50, 100, 200 and 300. The results are shown in fig.1.

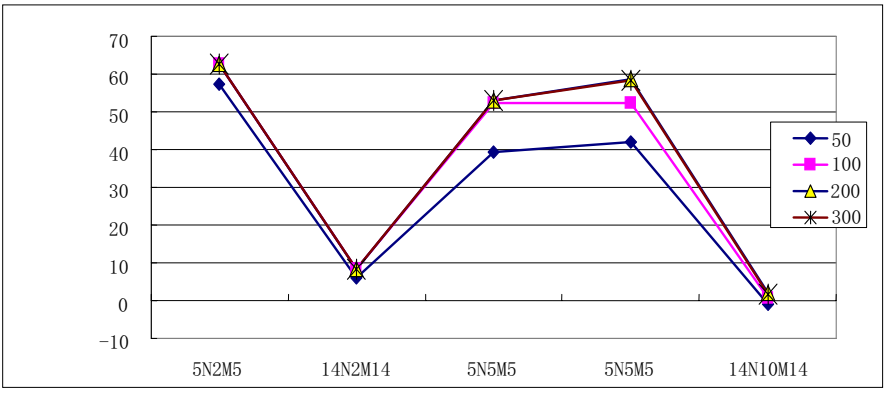


Fig. 1. Changing the population size

It can be concluded from fig.1 that when the population size is small, the result of the DSSSSEA algorithm is not good, when the population size is added to some extend such as 100, the efficiency of DSSSSEA algorithm improve quickly, but when the population size is still added, the efficiency of DSSSSEA algorithm has no more improvement. This is decided by the diversity of dynamic function optimization, the small population of course induce poor diversity of the population, but it don't mean that the larger population size induce better results, when the size is added to some extend, the population size is no more an effective fact for the results.

2) Changing the niche size
For convenience, we set the niche size as 50,100,200 and 300, and the population size is set as 100. The results are shown in fig.2.

It can be concluded from fig.2 that the change of niche size has almost no effect on the efficiency of the algorithm, no less which kind of change and which complex function. It is because that we use self-adaptive niche, that is to say, the better individual produce fewer children, on the other hand, the worse individual produce more children, at this time, the number of individuals in the niche is not so important.

3) Changing the memory size
To introduce the memory strategy is to record history optima to help latter searching in the related region. In the former experiments, we don't use memory strategy, so

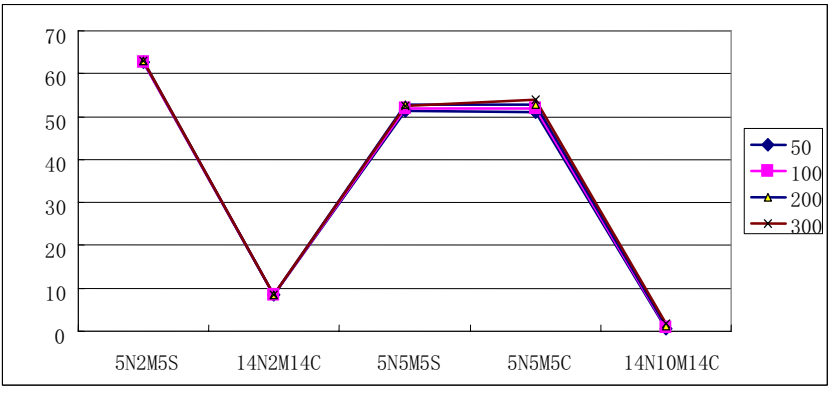


Fig. 2. Changing the niche size

how the memory strategy affects the algorithm? Let's ask for the answer in the following experiments. We reserve optima before change of condition every time, the memory initial set is the initial population. The results are shown in fig.3 and fig.4, where the percent means what percent of the population are replaced with the memory every time when the condition changes.

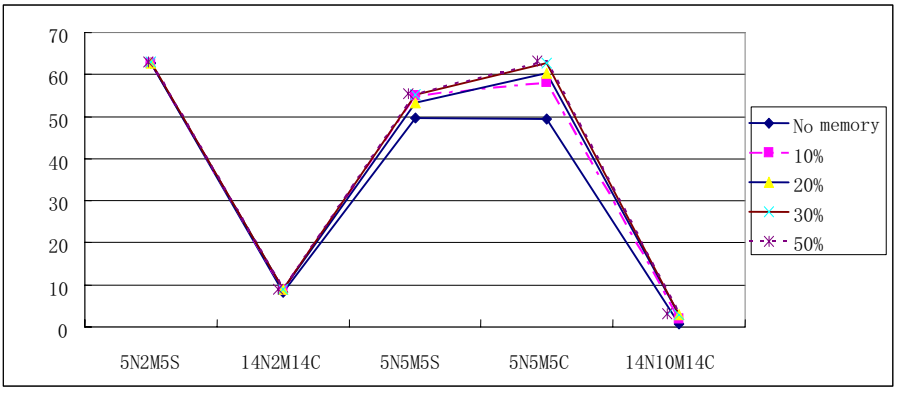


Fig. 3. Changing the memory

It can be concluded from fig.3 that the results of all test functions are better than those without memory, especially the percent from 0 to 10. However, for the low dimension function 5N2M1S and 14N2M14C, the effect on the algorithm with or without memory even the replacement percent with memory has no more than the high dimension function 5N5M5C and 5N5M5L, even there is a little improvement. As for the high dimension function 14N10M14C, since the values of its results are small, so we show its results in fig.4. From fig.4, we can see that the effect of memory on the results is very apparent. So, for the high dimension functions, we should set a memory to reserve history solutions, this is because when the complexity of the

function become higher, the history solutions can improve convergence speed. According to our experiments, for the percent of population replaced by memory, from 20% to 30% is a good choice.

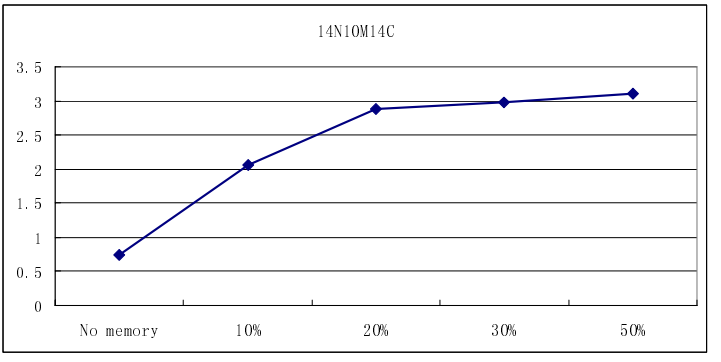


Fig. 4. Changing memory for 14N10M14C

4) Contrast with other algorithms

The change frequency is an important factor in the dynamic description. When we change the generation between two changes, the characteristics of the algorithm changes. For convenience, all of the algorithms for contrast don't use the strategy of memory, and the population size and niche size are all 100. At the same time, we set the mutation rate 0.3 in Hyper Mutation algorithm (it changes to 0.5 when the function changes), and set the percent of population replaced by randomly produced individuals 50% in Random Immigration algorithm. In SSSEA, we only use multi-parent crossover operator, the results are shown in fig.5.

From fig.5 we can see that for the five test functions of all kinds, the performance of DSSSEA is much better than the other three algorithms.

When the dimension of the dynamic function is low such as 5N2M1S and 14N2M14C in fig.5, the *Acc* values of DSSSEA, Hyper Mutation algorithm and

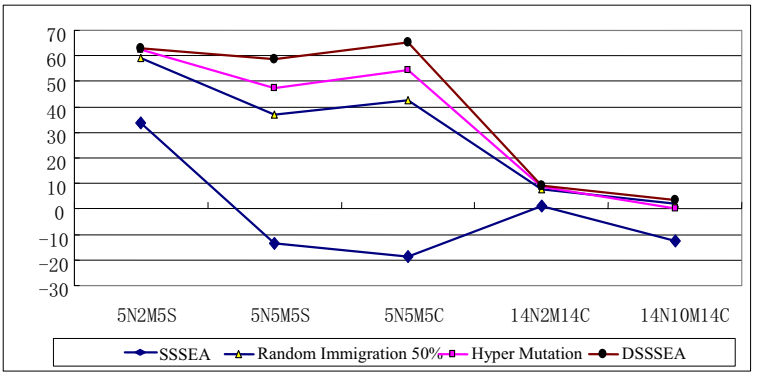


Fig. 5. Contrast with other algorithms

Random Immigration algorithm almost are the same point, but they are all better than SSSEA which only uses crossover operator, for higher dimension, the situation is more apparent. So we can conclude that performance of the mutation operator is much better than crossover operator in dynamic function optimization problems.

For higher dimension functions, such as 5N5M5C, 5N5M5L and 14N10M14C shown in fig.5, the order of *Acc* is:

Acc (Random Immigration) <*Acc* (Hyper Mutation) <*Acc* (DSSSEA)

So the order of the performance of these algorithms is:

Random Immigration < Hyper Mutation< DSSSEA

In fact, DSSSEA looks like Random Immigration and Hyper Mutation, but it uses self-adaptive mutation operator such that programming becomes easier without setting all kinds of parameters, at the same time, preserving the diversity of individuals in population concealed, which preserves good performance in all kinds of functions with all kinds of complexity or changing style.

4 Conclusions

According to the results of the above experiments, we can conclude that:

1) When the population size is small, the performance of DSSSEA is not so good, when the population size is added to some extend such as 100, the performance of DSSSEA improves quickly, but if continuing to add the size, the performance of DSSSEA has no more improvement.

2) Adding the niche size has no impressive improvement to the performance of DSSSEA.

3) For lower dimension function, with or without memory strategy and how many is the percent of population replaced by memory has no impressive improvement to the performance of DSSSEA, on the other hand, for higher dimension function, the percent of population replaced by memory has great effect on the performance of DSSSEA.

4) With the same parameter setting, the performance of DSSSEA is much better than those of Hyper Mutation algorithm, Random Immigration algorithm and SSSEA which only uses crossover operator.

Our experiments are simulated on an PC. So these conclusions on DSSSEA remain to be confirmed by using a distributed parallel computer to do more numerical experiments.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No.60473081) and the Natural Science Foundation of Hubei Province(No. 2005ABA234). Thanks especially give to the anonymous reviewers for their valuable comments.

References

1. Tan, K.C., Goh, C.K.: Handing uncertainties in evolutionary multi-objective optimization. In: Zurada, J.M., Yen, G.G., Wang, J. (eds.) Computational Intelligence: Research Frontiers. LNCS, vol. 5050, pp. 262–292. Springer, Heidelberg (2008)

2. Grefenstette, J.J.: Genetic algorithms for changing environments. In: Maenner, R., Manderichk, B. (eds.) Parallel Problem Solving from Nature2, pp. 137–144. North-Holland, Amsterdam (1992)
3. Cobb, H.G.: An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. Technical Report AIC-90-001, Naval Research Laboratory, Washington, USA (1990)
4. Grefenstette, J.J.: Evolvability in Dynamic Fitness Landscapes: A Genetic Algorithm Approach. In: Proceedings of the Congress on Evolutionary Computation, CEC 1999, pp. 2031–2038. IEEE, Los Alamitos (1999)
5. Deb, K., Udaya Bhaskara Rao, N., Karthik, S.: Dynamic Multi-Objective Optimization and Decision-Making Using Modified NSGA II: A Case Study on Hydro-Thermal Power Scheduling. Kanpur Genetic Algorithm Lab(KanGAL), Indian Institute of Technology, Kanpur, India, Technical Report 2006008 (2006)
6. Bosman, P.A.N., Poutre, H.L.: Learning and Anticipation in Online Dynamic Optimization with Evolutionary Algorithms: The Stochastic Case. In: Proceedings of the Congress on Evolutionary Computation, CEC 2007, pp. 1165–1172. IEEE Press, Los Alamitos (2007)
7. Yu, X., Tang, K., Yao, X.: An immigrants scheme based on environmental information for genetic algorithms in changing environments. In: 2008 IEEE World Congress on Computational intelligence (CEC 2008), pp. 1141–1147. IEEE Press, Los Alamitos (2008)
8. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Dordrecht (2002)

Improved GuoTao Algorithm for Unconstrained Optimization Problems

Ziyi Chen[1], Lishan Kang[1], and Lijun Liu[2]

[1] School of Computer, China University of Geoscience, Wuhan 430074, Hubei, China
chenziyi@263.net, kang_whu@yahoo.com
[2] Library, China University of Geoscience, Wuhan 430074, Hubei, China
chenliulijuan@yahoo.com

Abstract. This paper presents an improved GuoTao algorithm for solving unconstrained optimization problems. The algorithm combines the multi-parent crossover operator of GuoTao algorithm with two local search operators. The first local search operator is a simplified Hooke-Jeeves algorithm which can find near-optimal solutions near the start point within reasonable computational time. And the second local search operator is a powell algorithm which can find the local optimal solution near the start point. The multi-parent crossover operator can enables individual to draw closer to each local optimal solution, thus the population will be divided into subpopulations automatically, meanwhile, the first local search operator is adopted to gain near-optimal solution in each subpopulation. As a result, the promising areas are gained quickly which decrease useless search greatly. At last, the second local search operator is adopted to find the global optimal solution only from those promising areas.Numerical experiments using a suite of test functions, which are widely studied in the field of evolutionary computation, show that the proposed algorithm is superior to GuoTao algorithm for solving unconstrained optimization problems with high dimensionality.

1 Introduction

Evolutionary computation has become an important problem solving methodology among many researchers. The population-based collective learning process, self adaptation,and robustness are some of the key features of evolutionary algorithms when compared to other global optimization techniques.Some of the popular approaches are real-parameter EAs, evolution strategies (ES), differential evolution (DE), particle swarm optimization (PSO), evolutionary programming (EP), classical methods such as quasi-Newton method (QN), hybrid evolutionary-classical methods, other non-evolutionary methods such as simulated annealing (SA), tabu search (TS)[1-5] and others[6].

In 1999,GuoTao presented a novel evolutionary algorithm [7] which has shown that it is a very effective algorithm for unconstrained optimization problems. The most important feature of GuoTao algorithm lies its multi-parent crossover operator.The multi-parent crossover operator can enables individual to draw closer to each local optimal solution, thus the population will be divided into subpopulations

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 37–44, 2008.
© Springer-Verlag Berlin Heidelberg 2008

automatically, all individuals in each subpopulation belong to a same attractor. At last, each attractor will be converged to a local optimal solution. By this way, GuoTao algorithm can find the global optimal solution with large probability. But there are two shortcomings in GuoTao algorithm. First of all, because GuoTao algorithm lacks local search operator, the convergence velocity of each subspace is relatively slow. Second, many subspaces are not promising areas, but the algorithm has to spend lots of time to search these subspaces. How to solve these two problems to improve the performance of GuoTao algorithm ?Generally, hybrid evolutionary algorithm is a good choice[8]. A simple idea is to add a local search operator to GuoTao algorithm, then each subspace can converge to a local optimal solution quickly. But this method can not avoid the second defect. How to find the promising areas [9]? So this paper presents an improved GuoTao algorithm for solving unconstrained optimization problems. The algorithm combines the multi-parent crossover operator of GuoTao algorithm with two local search operators. The first local search operator is a modified Hooke-Jeeves algorithm[10]. In fact, it is a incomplete local search operator which can find near-optimal solution near the start point, its goal is not to find the local optimal solution in the subspace. The function of the first local search operator is to find those promising areas with little computation complexity, so there is no need to carry out the complete local search which is a time consuming process.As a result, the promising areas are gained quickly which decrease useless search greatly. At last, the second local search operator is adopted to find the global optimal solution only from those promising areas.In fact, the second local search operator is a powell algorithm which can find the local optimal solution near the start point[11].

When the algorithm run, the multi-parent crossover operator can enables individual to draw closer to each local optimal solution, thus the population will be divided into subpopulations automatically,meanwhile, the first local search operator is adopted to gain near-optimal solution in each subpopulation. As a result, the promising areas are gained quickly which decrease useless search greatly. At last, the second local search operator is adopted to find the global optimal solution only from those promising areas.Numerical experiments using a suite of test functions, which are widely studied in the field of evolutionary computation, show that the proposed algorithm is superior to GuoTao algorithm for solving unconstrained optimization problems.

The rest of this paper is organized as follows: Section 2 gives the structure of proposed algorithm(LGT) and describes the implementation of the algorithm. The experiments and comparisons with GuoTao algorithm are illustrated by solving several benchmark problems in section 3.Finally,section 4 concludes with some brief remarks.

2 Descriptions of Improved GuoTao Algorithm (LGT)

The algorithmic description of LGT is as follows:

```
PROCEDURE LGT ALGORITHM
BEGIN
Randomly initialize population  P={ X_1, X_2,..., X_N };
  generation=0;  t=0 ;
```

```
while(the termination condition is not satisfied) do
BEGIN
  select M points X'_1, X'_2,..., X'_M  from population ;
    X=GTX( X'_1, X'_2,..., X'_M ) ;
  IF  better(X, X_worst)  THEN BEGIN
                X_worst = X;
                X= FirstLocalSearch(X);
                IF  better(X, X_worst)  THEN BEGIN
                   X =SecondLocalSearch(X);
                     X_worst = X;
                END ELSE BEGIN
                  X_worst = X;
                  END;
      END;
      generation = generation + 1;
    END;
    Output Result;
 END;

END.
```

2.1 Multi-parent Crossover Operator(GTX) of GaoTao Algorithm

In GuoTao algorithm, a linear non-convex multi-parent crossover operator (GTX) was proposed to generate a child.

Its algorithm is as follows:

Suppose $a_1^{t-1}, a_2^{t-1}, \ldots, a_m^{t-1}$ are m parents in the *t-1* generation with $\lambda_1, \lambda_2, \ldots, \lambda_m$ as their crossover coefficient respectively. a^t is the child in the t generation, which is obtained according to the following formula:

$$a^t = \lambda_1 * a_1^{t-1} + \lambda_2 * a_2^{t-1} + \ldots + \lambda_m * a_m^{t-1}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_m \in [-0.5, 1.5]$ and $\lambda_1 + \lambda_2 + \ldots + \lambda_m = 1$

Linearity enables the offspring to inherit possible linkages among parameters. Non-convexity tends to guide the population to search areas outside where it resides. Simplicity and high efficiency are the main advantages of GTX. However there are two main shortcomings in GuoTao algorithm. First, because GuoTao algorithm lacks local search operator, the convergence velocity of each subspace is relatively slow. Second, many subspaces are not promising areas, but the algorithm has to spend lots of time to search these subspaces. So GuoTao algorithm often meet with trouble when solving the large scale function optimization problems.

2.2 First Local Search Operator

The first local search operator is a modified Hooke-Jeeves algorithm.The pattern search method of Hooke and Jeeves is a sequential technique, each step of which consists of two kinds of moves, the exploratory move and the pattern move. The first kind of move is included to explore the local behaviour of the objective function and the second kind of move is included to take advantage of the pattern direction. In fact, the pattern move is only an accelerative method.If we remove the pattern move, the algorithm is still a local optimization algorithm, of course, the velocity of convergence of the modified algorithm will be slower than the origin algorithm. But the modified algorithm has a important merit that the search depth can be controlled according to the initialization parameter.

The general procedure of the modified algorithm can be described by the following steps:

1. Start with an arbitrarily chosen point called the starting base point X_1, and prescribed step lengths Δx_i in each of the coordinate directions $\mathbf{u}_i$, *i=1,2,..,n.* Set *k=1.*
2. Compute $f_k = f(\mathbf{X}_k)$. Set *i=1,* $\mathbf{Y}_{k0}=\mathbf{X}_k$, where the point $\mathbf{Y}_{kj}$ indicates the temporary base point $\mathbf{X}_k$ by perturbing the *j*th component of $\mathbf{X}_k$. Then start the exploratory move as stated in Step 3.
3. The variable x_i is perturbed about the current temporary base point $\mathbf{Y}_{k,i-1}$ to obtain the new temporary base point as:

$$\mathbf{Y}_{k,i}=\begin{cases}\mathbf{Y}_{k,i-1}+\Delta x_i\mathbf{u}_i & \text{if } f^+=f(\mathbf{Y}_{k,i-1}+\Delta x_i\mathbf{u}_i)<f=f(\mathbf{Y}_{k,i-1})\\ \mathbf{Y}_{k,i-1}-\Delta x_i\mathbf{u}_i & \text{if } f^-=f(\mathbf{Y}_{k,i-1}-\Delta x_i\mathbf{u}_i)<f=f(\mathbf{Y}_{k,i-1})<f^+=f(\mathbf{Y}_{k,i-1}+\Delta x_i\mathbf{u}_i)\\ \mathbf{Y}_{k,i-1} & \text{if } f=f(\mathbf{Y}_{k,i-1})<\min(f^+,f^-)\end{cases}$$

   This process of finding the new temporary base point is continued for *i=1,2,...* until x_n is perturbed to find $\mathbf{Y}_{k,n}$.
4. If the point $\mathbf{Y}_{k,n}$ remains the same as $\mathbf{X}_k$, reduce the step lengths Δx_i (say, by a factor of 2), set *i=1* and go to step 3. If $\mathbf{Y}_{k,n}$ is different from $\mathbf{X}_k$, obtain the new base point as :

$$\mathbf{X}_{k+1}=\mathbf{Y}_{k,n}$$

   and go to step 5.

5 Set *k=k+1,* $f_k=f(\mathbf{Y}_{k0})$, *i=1*, and repeat step 3. If at the end of step 3, $f(\mathbf{Y}_{k,n}) < f(\mathbf{X}_k)$, we take the new base point $\mathbf{X}_{k+1}=\mathbf{Y}_{k,n}$ and go to step 5. On the other hand, if $f(\mathbf{Y}_{k,n}) \geq f(\mathbf{X}_k)$, set $\mathbf{X}_{k+1}\cong\mathbf{X}_k$, reduce the step lengths Δx_i, set *k=k+1*, and go to step 2.

6 The process is assumed to have converged whenever the step lengths fall below a small quantity ε. Thus the process is terminated if

$$\max_i(\Delta x_i)<\varepsilon$$

2.3 Second Local Search Operator

To avoid the disadvantages of the traditional optimization algorithms, we adopt Powell algorithm to implement the complete local search algorithm, because Powell algorithm belongs to the class of direct search methods, a class of optimization algorithms which neither compute nor approximate any derivatives of the objective function.

3 Numerical Experiments and Analysis

In this section, we choose five typical global minimization problems and give the comparisons to other results. These examples are often used to test the performance and measure the reliability of a global optimal algorithm.

When testing the algorithm on well understood problems,there are two measures of performance:

1. Success rate: when the algorithm found the global optimal solution, we think this run as a success run, so the success rate is the ratio of total success runs and total runs.
2. The average number of objective function evaluations required to find these optima.

Example 1. De Jong function [12]

$$f(\vec{x}) = \sum_{i=1}^{n} x_i^2 \qquad -5.12 \le x_i \le 5.12$$

The optimal solution is the origin of the coordinate system.

Example 2. Rosenbrock function [12]

$$f(\vec{x}) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad -2.048 \le x_i \le 2.048$$

It is unimodal but has strong epistasis among parameters.The optimal solution is (1,1...1), which is located on a curved valley.

Example 3. Griwank function [13]

$$f(\vec{x}) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} (\cos(x_i / \sqrt{i})) \quad -512 \le x_i \le 512$$

The optimal solution is the origin of the coordinate system.There are many local minima around the the global minimum.

Example 4. Rastrigin function [14]

$$f(\vec{x}) = nA + \sum_{i=1}^{n} (x_i^2 - A\cos(2\pi x_i)) \quad -5.12 \le x_i \le 5.12$$

The optimal solution is the origin of the coordinate system.There are many local minima around the global minimum.

Example 5. Schwefel function [3]

$$f(\vec{x}) = -\sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|}) \quad -500 \le x_i \le 500$$

The global minimum is $x_i = 420.9687$, i =1,2,..,n.

3.1 The Comparison of Proposed Algoritm(LGT) and the Algorithms GuoTao

To compare the performance of the proposed algorithm(LGT) with the algorithm GuoTao, the same parameters are adopted in algorithm LGT and GuoTao for each test problem. Table 1 and Table 2 summarizes the experimental results we obtained by using LGT and GuoTao and statistics for the 30 independent runs.

Control parameters are set as follows::dimension(n),poplation size(N), the number of the parents in multi-parent crossover operator(M).

1. When the problems have low dimensionality: n=20, N =100, M=10;
2. When the problems have high dimensionality: n=100,N=1000, M=10.

Table 1. The comparison of proposed algorithm(LGT) and the algorithms GuoTao (when the dimension n=20)

Example No	Algorithm	Dimensions	Success rate	Average number of function evaluations
Example 1	GuoTao	20	100%	366890
	LGT	20	100%	2567824
Example 2	GuoTao	20	100%	652336
	LGT	20	100%	5468462
Example 3	GuoTao	20	100%	853090
	LGT	20	100%	6875464
Example 4	GuoTao	20	100%	374902
	LGT	20	100%	2839462
Example 5	GuoTao	20	100%	486438
	LGT	20	100%	6948672

From the Table 1, we can see that LGT and GuoTao are both surprisingly good for solving the function optimization problems with low dimensionality, they can find the optimal solution in every run for all benchmark problems. Note that the average number of function evaluations of LGT is larger than that of the algorithm GuoTao for every test problems, the reason lies that these low dimension problems are relatively simple to the algorithm GuoTao, there is no need to adopt local search operator to find the global optimal solution.

Table 2. The comparison of proposed algorithm(LGT) and the algorithms GuoTao (when the dimension n=100)

Example No	Algorithm	Dimen-sions	Success rate	Average number of function evaluations
Example 1	GuoTao	100	0%	
	LGT	100	100%	468261836
Example 2	GuoTao	100	0%	
	LGT	100	70%	843268642
Example 3	GuoTao	100	0%	
	LGT	100	100%	624850338
Example 4	GuoTao	100	0%	
	LGT	100	83%	532788504
Example 5	GuoTao	100	0%	
	LGT	100	90%	956796226

From the Table 2, we can see that LGT is still surprisingly good for solving the function optimization problems with high dimensionality, but GuoTao will meet with trouble for such problems because it lacks effective local search operator.

4 Conclusions

This paper presents an improved GuoTao algorithm for solving unconstrained optimization problems. The algorithm combines the multi-parent crossover operator of GuoTao algorithm with two local search operators. The multi-parent crossover operator can divided the population into subpopulations automatically, meanwhile, the first local search operator is adopted to gain near-optimal solution in each subpopulation. As a result, the promising areas are gained quickly, then the second local search operator is adopted to find the global optimal solution only from those promising areas. Numerical experiments using a suite of test functions, which are widely studied in the field of evolutionary computation, show that the proposed algorithm is surprisingly good for solving unconstrained optimization problems, specially the proposed algorithm is superior to GuoTao algorithm when the optimization problem has high dimensionality.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. Univ. of Michigan Press, Ann Arbor (1975)
2. Rechenberg, I.: Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog, Stuttgart (1973)
3. Schwefel, H.P.: Numerical Optimization of Computer Models. John Wiley, Chichester (1981)

4. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley, New York (1966)
5. Koza, J.R.: Genetic Programming: on the programming of computers by means of natural election. MIT Press, Cambridge (1992)
6. Yao, X., Xu, Y.: Recent Advances in Evolutionary Computation. Journal of Computer Science and Technology 21(1), 1–18 (2006)
7. Tao, G., Li-shan, K.: A New Evolutionary Algorithm for Function Optimization. Wuhan University Journal of Nature Sciences 4(4), 409–414 (1999)
8. Grosan, C., Abraham, A.: Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews. Studies in Computational Intelligence (SCI) 75, 1–17 (2007)
9. Oliveira, A.C.M., Lorena, L.A.N.: Detecting promising areas by evolutionary clustering search. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS, vol. 3171, pp. 385–394. Springer, Heidelberg (2004)
10. Walsh, G.R.: Methods of Optimization. John Wiley and Sons, London (1975)
11. Findeisen, W., Szymanowski, J., Wierzbicki, A.: Teoria i metody obliczeniowe optymalizacji. PWN, Warszawa (1980)
12. De Jong, K.A.: An Analysis of Behavior of a Class of Genetic Adaptive Systems. Ph.D. thesis, University of Michigan, Ann Arbor, Michigan (1975)
13. Griewangk, A.O.: Generalized Descent for Global Optimization. JOTA 34, 11–39 (1981)
14. Pan, Z.J., Kang, L.S., Chen, Y.P.: Evolutionary Computation, pp. 67–68. Tsinghua University Press, Beijing (1998)

Gene Expression Programming Based on Subexpression Library and Clonal Selection

Siqing Xue and Jie Wu

School of Computer Science, China University of Geosciences,
430074 Wuhan, China
xuesiqing@gmail.com

Abstract. Gene expression programming (GEP) is a recently developed evolutionary computation method for model learning and knowledge discovery. Sometimes it is not easy when use GEP to solve too complex problem, so enhancing the algorithm learning capability is necessary. This paper proposes an immune principle based GEP algorithm (iGEP), which combines gene library and clonal selection algorithm. The gene library is composed of subexpressions of GEP expression selected from the process of evolution. The proposed algorithm introduces some new features, including the best subexpression of GEP expression is selected as the solution of the problem, and some segments of gene library are used for hypermutation and receptor editing. In terms of convergence rate and computational efficiency, the experimented results on some benchmark problems of the UCI repository show that iGEP outperforms the standard GEP.

Keywords: gene expression programming, clonal selection, gene library, best subexpression.

1 Introduction

Gene expression programming [1] is a recently developed evolutionary computation method for model learning and knowledge discovery. Because of its capability of using linear chromosome to represent the nonlinear computer program, GEP dramatically improves on traditional genetic programming (GP) [1] in the term of complexity and time efficiency. Previous research work has shown its powerful capabilities over a large range of domains, including regression modeling, classification tasks, time series prediction, and logic synthesis [1][2][3][8][9][10]. Despite the initial success of GEP, most of the work addressed the application of GEP, and there remain many open issues to deal with, including learning capability research, theoretical analysis, real-world applications, etc. In this paper we will focus on enhancing the GEP algorithm learning capability based on immune principle.

Over the last few years, the study of artificial immune systems (AIS) has become increasingly important due to the large number of its possible applications in fields of science and engineering [4]. Based on the learning and evolutionary principle in the adaptive immune response, De Castro developed CLONal selection ALGorithm (CLONALG) [4], which uses individual mutation to perform greed search and uses random search to explore the whole solution space. CLONALG emphasizes proportion selection and clone, and the essence of the clonal operator is producing a

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 45–57, 2008.
© Springer-Verlag Berlin Heidelberg 2008

variation population around the parents according to their affinity, and then the searching area is enlarged. Compared with standard genetic algorithm (GA), CLONALG is convergent faster and the diversity is much better. But when the objective function becomes complex, the search ability of the algorithms becomes weak. Although the algorithms proposed in Ref. [4] try to simulate the characters of the diversity of population, the CLONALG emphasizes the more local search. On the other hand, Gene libraries are a biological mechanism for generating combinatorial diversity in the immune system [5][6]. In this paper, we will adopt the clonal selection with hypermutation and GEP subexpression library as the search strategy of the immune principle based Gene Expression Programming (iGEP) algorithm.

In the next section of this paper, a brief review of related work and the motivation of the algorithm design are presented, and then a detailed description of the iGEP algorithm is put forward in Section 3. In Section 4, numerical experiments are performed on a suite of benchmark functions to investigate the effectiveness of the proposed algorithm. Finally, some conclusions are presented in Section 5.

2 Related Work and Motivation

2.1 Gene Expression Programming

In order to describe the problem-solving paradigm of GEP we focus our overview on symbolic regression problems. Like GP, suppose the function set, terminal set, fitness function, control parameters and stop condition have been determined. GEP algorithm, which is in general no different from the traditional GP, begins with random generation of the initial population. Then individuals are evaluated (after being translated into algebraic expression) and selected accordingly to the fitness to reproduce with mutation or crossover operation. Individuals of the new generation will be evaluated, selected and then reproduced with evolution operation to generate the next one. The above process will be repeated for a certain number of generation or until a solution has been found. The most difference between GEP with traditional GP is the expression of the chromosomes (i.e. individuals), and therefore the operators on the genes and the evaluation process of the individuals.

1) GEP Genes

Unlike the parse tree representation in traditional GP, GEP uses a fixed string (i.e. chromosome or GEP gene) to represent the solution of problem. Each gene is composed of a head and a tail, where the head contains the function symbols or terminal symbols but the tail contains only terminals. For a given problem, when the length of the head h is chosen, the length of the tail t is evaluated by equation (1), where n is the number of arguments of the function with more arguments .

$$t = h\,(n-1) + 1\ . \tag{1}$$

The gene will be mapped into an expression tree (ET) following the width-first fashion. For example, given the function set {+, -, *, /, S} and the terminal set {a, b, c}. In this case, n=2; and h=8, then t=9. So the length of the gene is 8+9=17. The following is a GEP gene:

$$\begin{array}{l} 0\,1\,2\,3\,4\,5\,6\,7\,8\,9\,0\,1\,2\,3\,4\,5\,6 \\ *.-./.S.c.*.c.+.\ \ a.b.a.b.a.a.a.b.b. \end{array} \tag{2}$$

Where '.' is used to separate function symbols for easy reading; S is the Sin () function; and a, b, c are variable names. The underlined substring of expression (2) is the head of the expression and the rest is the tail and the substring with grey shading is K-expression of this expression.

The above string with grey shading, named K-expression [1], is the valid segment of the chromosome, and can be represented as an ET (Fig 1). K-expression of expression (2) is the straightforward reading of the ET from left to right and from top to bottom. Following the in-order fashion, the above ET can be mapped into the algebraic expression (3), i.e. the phenotype of the chromosome, which will be used to evaluate the fitness of the gene.

$$(\mathrm{Sin}\,(a + b) - c)*((a * b) / c)\,. \tag{3}$$

It can be proved that the algebraic expression (3) can be, very straightforward, converted into the ET (Fig 1) and then into the K-expression of gene expression (2). The genetic code is very simple: a one-to-one relationship between the K-expression of the chromosome and the solution of problem.

2) GEP Operators

In GEP, the roulette-wheel sampling and the elitism selection are chosen as the selection strategy, which can guarantee the best individuals being cloned to the next generation. Unlike other GP algorithm, GEP creates three specific modification operators corresponding to the K-expression, which are recombination, mutation and transposition [1].

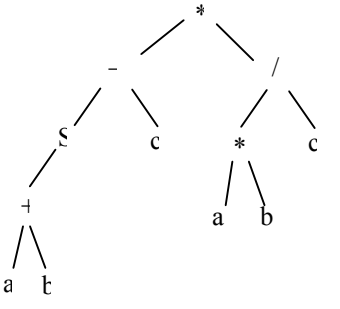


Fig. 1. Expression tree of expression (2)

1) Recombination. There are three kinds of operators: one-point, two-point and gene recombination, which work in the same fashion as that in GA.

2) Mutation. According to the definition of the specific K-expression, in the heads, any symbol can change into function or terminal, but in the tails, the symbol can only change into terminal.

3) Transposition. Any fragment of the genome can transpose to a randomly chosen point.

Since having the especial structure of gene and operators as mentioned above, GEP is convenient and efficient implementation due to its linear genotype as GA, and capable of retaining a certain extent of functional complexity due to its phenotype as GP.

With the flexibility, capability and efficiency, GEP has been widely applied to the field of the intelligent computation, including symbolic regression, classification rules discovery and machine learning, etc.

3) Learning Capability of GEP

Sometimes it is not easy when use GEP to solve too complex problem, so enhancing the algorithm learning capability is necessary. In ref. [1], C. Ferreira introduced Automatically Defined Functions into GEP, and ref. [7] divided the head of chromosomes into a head and a body and designed a multi-layer chromosomes approach, named M-GEP. Authors in ref. [8] implemented a hybrid GEP algorithm based on CLONALG. Ref. [9][10] proposed a theory about self-emergence of structures in

GEP, which means structures are generated, preserved, and evolved automatically during the evolutionary process. This approach, called P-GEP, is based on a new linear genotype representation in prefix notation instead of K-expression.

2.2 Motivation

1) Evolvability. GEP has been used to solve a huge variety of problems, however, in the term of evolvability (a measure of an evolutionary system's ability to evolve in an appropriate direction [11][12]) and biological systems, GEP is far from perfect. Author in ref. [12] introduced a few important approaches to improving the evolvability of evolutionary computation techniques, such as pleiotropy, redundancy, and positional independence. Redundancy provides structural protection from change and also plays a range of interesting roles in GEP, which is the main reason that GEP satisfies the operation closure and so the efficiency is higher than GP. However, it is hard to introduce pleiotropy to GEP and to implement the condition of positional independence since the mapping mechanism of genotype-phenotype. And also taking inspiration from human immune system, in this paper we consider the approach to improve the evolvability of GEP based on AIS and substructures learning.

2) Best Subexpression. In the standard GP or GEP only on variable is chosen to provide the output of the problem, but in fact each subexpression (subtree) may be considered as a potential solution of the problem, especially when the complexity of the target problem is not known. The Multi-Solution ability has been tested within other evolutionary model such as Multi Expression Programming [13][14].

GEP chromosomes are usually composed of multiple genes. Each gene codes a sub-ET and the sub-ETs are linked by a function. But it is not a good idea to assume that the genes may be linked either by addition or by multiplication [13].

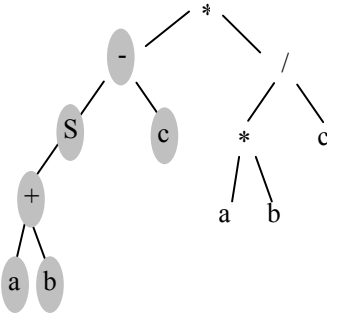


Fig. 2. Best subexpression tree

We know that a GEP gene, i.e. GEP expression (encoded into expression tree) includes a few subexpressions (encoded by the subtrees of the original tree). If each of subexpressions may be chosen to represent the GEP expression, we may say that the GEP expression has a manifold phenotypic transcription [13], which gives a supplementary power to GEP method. For instance, expression (2) is a GEP gene, which corresponding expression tree is illustrated in Fig 1 and algebraic expression is expression (3). We may find that the grey filled nodes form the best fitted expression encoded in this tree (called best subexpression). In fact, the subexpression of expression (3) encoded by the grey subtree is properly the target solution (see Fig 2).

3) Clonal Selection Principle and Gene Library. Artificial Immune Systems (AIS) have been shown to be useful, practical and realizable approaches to real-world problems. Three major evolution processes involved in the human immune system are gene library evolution, negative selection and clonal selection [6].

The CONSALG algorithm developed on the basis of clonal selection theory [4], which main idea is that only those immune cells that recognize the antigens are selected to proliferate. The antigens can selectively react to the antibodies, which are the natural production and spread on the cell surface in the form of peptides. The reaction leads to cell proliferating clonally and the clone has the same antibodies. The clonal selection is a dynamic process of the immune system self-adapting against antigen stimulation, which has some biologic characters such as learning, memory and genetic diversity.

Gene libraries are a biological mechanism for generating combinatorial diversity in the immune system. However, they also bias the antibody creation process, so that they can be viewed as a way of guiding lifetime learning mechanisms. The related work introduced in the previous section indicates two important strategies of the human immune system: (1) obtaining the germ line diversity through gene library evolution is somewhat directed toward covering a coarse-grained antigen space, and (2) learning through hypermutation leads the immune system to fine-tune its detection of the existing antigens [6].

Thus, combining clonal selection with hypermutation, gene library and best subexpression introduced above, of which modularity and implicit reuse are two features, will be used to introduce pleiotropy to GEP, and to some extent to enhance the positional independence of GEP genotype-phenotype mapping.

3 Proposed Approach

The iGEP algorithm, introduced in this paper, is a novel paradigm combining immune principles of clonal selection and gene library with GEP representation of solutions to problems. The algorithm is briefly described as follows.

Step 1. [Initialization] An initial population, P, of N GEP expressions (antibodies), is randomly generated.

Step 2. [Evaluation] An antigen, Ag, is presented to the population and for each antibody the affinity is determined.

Step 3. [Library update] With a certain probability, p_l, the best N * p_l antibodies, in terms of their affinity, are selected to update a subexpression gene library.

Step 4. [Receptor editing] With a certain editing probability, p_e, the N * (1- p_c) * p_e antibodies with lower affinities are selected randomly to undergo receptor editing. The set of receptor edited is called EP, where p_c is the clone probability.

Step 5. [Cloning] The best n_c = N * p_c antibodies are selected and each is cloned independently and proportionally to their affinity to form n_c subpopulation.

Step 6. [Hybrid hypermutation] Each clone of each subpopulation is submitted for hybrid hypermutation with a probability inversely proportional to its antigenic affinity. The mutated clones form n_c subpopulation.

Step 7. [Affinity of Clones] The affinity is computed for each mutated clone and the highest affinity individual in each subpopulation is selected to compose a new population named MP. If the affinity of the best individual in subpopulation is lower than cloned antibody, then the cloned one will be selected into new population.

Step 8. [Population refresh] With a certain probability, p_r, the N * p_r new antibody is generated to form a new population named NP, which will replace the lowest affinity individuals in the popular P.

Step 9. [Algorithm iteration] Calculate the affinity of each individual in popular EP and NP; refresh popular P with EP, MP and NP and the new population is submitted to *step 3*. The process continues iteratively until a stopping criterion is met.

1) Affinity Assignment. Computing the fitness of the GEP expression based on the best subexpression method is an easy and inexpensive task. We know that the process of determining the fitness of GEP expression is the process of K-expression is computed, i.e., translating K-expression into a expression tree rooted by the first non-terminal symbol of the K-expression and then computing the algebraic expression encoded by this expression tree. Obviously, the values of all the subexpression of the algebraic expressions (encoded into by the subtrees) are also computed, which means that computing the value of an expression and of all subexpressions requires the same number of operations as the computation of the value of the entire expression. According the mapping of K-expression between subexpression, we employ a bottom-up and right-left approach. Let's consider the K-expression encoded by tree in Fig 2. The order for fitness computation is the following:

1: + (a, b)
2: * (a, b)
3: S (2)
4: / (2, c)
5: - (3, c)
6: * (4, 5)

The last expression whose fitness is computed is the one encoded by the entire tree. In order to get the fitness of each subexpression we only have to take the difference (in absolute value) between the actual values of subexpression and expected output and then we sum these results for all cases. The fitness of each subexpression *subExp[i]* of GEP expression *Exp* may be computed using the following formula:

$$f(subExp[i]) = \sum_{j=1}^{m} | s_{i,j} - d_j | \ . \tag{4}$$

Where $s_{i,j}$ is the result obtained in subexpression *subExp[i]* for the fitness case *j*, d_j is the targeted result for the fitness case *j* and *m* is the number of fitness cases.

For symbolic regression problem the fitness needs to be minimized. The fitness of an *Exp* is set to be equal to the lowest fitness of the subexpression, i.e., the best subexpression in GEP expression *Exp* (see equation(5)).

$$f(Exp) = \min_i (f(subExp[i])) \ . \tag{5}$$

2) Subexpression Library. The subexpression library maintains the substructures of the best subexpression. The library is more like dynamic substructure library [8], but there are several important features:

(i). The library is a collection of previously known best substructures which is the subexpressions of best subexpression of selected GEP expression. The size of library is an important parameter defining the subexpression library.
(ii). Through hypermutation and receptor editing, evolution play an important role in maintaining the learnt behavior and the diversity of the library (see Fig 3).

(iii). The subexpression is randomly selected for hypermutation or receptor editing according to the required complexity (length).
(iv). The lower affinity, the bigger size the subexpression in library is, the more probability is to be updated by new subexpression.
(v). All the unrepeated subexpression will be stored in form of K-expression.

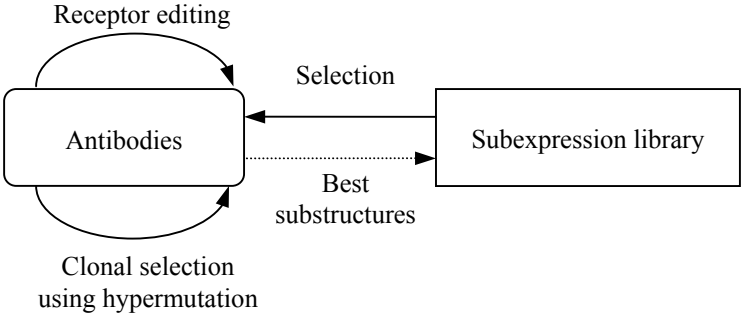


Fig. 3. Subexpression library in iGEP

3) Receptor Editing. The receptor editing process is to form new antibodies from random subexpressions of gene library. Those antibodies will replace the lowest affinity antibodies in order to enhance the learning ability of algorithm, and also improve the diversity of the population. The following algorithm described the process:

Step 1. [Initialization] Select randomly a function as the first symbol of the antibody Ab, l_a=1.
Step 2. [Subexpression selection] A subexpression, which complexity l_s is less than the h - l_a, is selected randomly from the gene library. Where h is the length of the head.

Step 3. [Formation of new antibody] The selected subexpression is appended to the Ab, thus l_a=l_a+l_s.

Step 4. [Algorithm iteration] If the current length of the antibody l_a is less than h, then go to *Step 2*. Otherwise, the algorithm terminates.

4) Cloning. The process of cloning and hypermutation of the antibody make the population evolve so that the diversity of the population is maintained and the local search ability of the algorithm is enhanced. The clonal proliferation is similar to that of immunology, which is a simple process of propagating asexually, so that a group of "good" genetically cells can be descended to the most possible extent. iGEP algorithm uses almost the same control mechanisms (given by Equation (1)) as CLONALG.

$$n_i = round\left(\frac{\beta \cdot n_c}{i}\right),\ 1 \le i \le n_c \tag{6}$$

Where, i is the index of current selected antibody, n_i is the number of clones generated, n_c is the total number of antibodies selected in Step 5 of the algorithm, and β is a constant, called clone factor, round(·) is the rounding function.

5) Hybrid Hypermutation. The antibodies generated by cloning are submitted to a hypermutation procedure. The inverse proportionality of hypermutation ensures that search space is expanded. But under the GEP genotype-phenotype mapping mechanism does not guarantee that the levels of functional complexity in the phenotype are also directly reflected in the genotype. Thus in order to enhance the learning ability and locality search ability of iGEP algorithm, we design the following two mutation operators.

Random mutating: using the same mutation strategy as GEP, the mutation process is implemented by replacing some symbols of antibodies (GEP expressions) generated by cloning with some new randomly generated symbols belonging to the defined function set or terminal set. This mutation maintained the diversity mechanism.

Subexpression mutation: this process will replace one symbol of an antibody with a subexpression (subtree) selected from expression library. To complete the mutation, the following rules are used.

(i). The selected symbol from antibody must be valid, that is, the symbol must be one of the responding K-expression. In addition, the first symbol of the head can not be selected to mutate.
(ii). Suppose that l_s is the number of non-terminal in selected subexpression and l_m is the length of the segment from mutated symbol to the end of the head. So l_s should be less than or equal to l_m.
(iii). The last l_s - 1 symbol of the head will be deleted.
(iv). The method of fitness computation of the offspring is the same as original GEP, but the subexpression look as a terminal, whose value is checked from caching subexpression.

To illustrate the process of hypermutation, let us consider expression (2) again. Assume that the mutation process selects symbol at position 2, 3, 5, and 6 to mutate and then yields the following four expressions (the left expression is original and the right is responding offspring):

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
.-./.S.c..c.+. a.b.a.b.a.a.a.b.b.	*.-.①①S.c.*. c. a.b.a.b.a.a.a.b.b.
.-./.S.c..c.+. a.b.a.b.a.a.a.b.b.	*.-./.①①c.*.c. a.b.a.b.a.a.a.b.b.
.-./.S.c..c.+. a.b.a.b.a.a.a.b.b.	*.-./.S.c. ①①c. a.b.a.b.a.a.a.b.b.
.-./.S.c..c.+. a.b.a.b.a.a.a.b.b.	*.-./.S.c.*. ①① a.b.a.b.a.a.a.b.b.

(7)

Where, the symbol with grey shading will be submitted for mutation, the underlined substring is head, and the digit is the number of subexpression selected from gene library. The resulting offspring can correspondingly be mapped into the expression tree shown in Fig 4. Grey filled nodes belong to the subtree mapped from the subexpression selected from gene library. It can be observed that the subtree from library is

preserved and transmitted into the offspring. In addition, the selected subtrees are more likely the subtree of the best subtree, thus we can conclude that the substructures of the best GEP expression are transmitted.

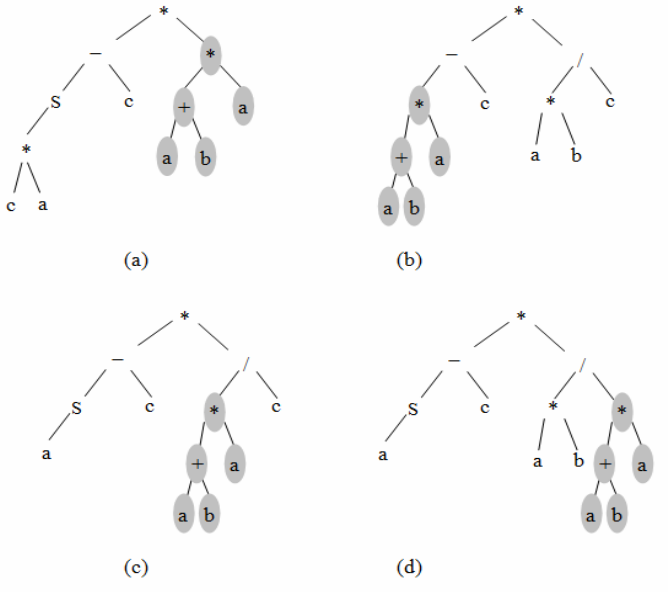


Fig. 4. Illustration of subexpression mutation

6) Caching Subexpression. One of the major drawbacks of Genetic Programming, including GEP, MEP and iGEP in this paper is the need for great computational power. On the other hand Hypermutation in iGEP algorithm requires repeated evaluation of the individuals too. When cloned antibodies are submitted for hypermutation, some function symbols or variables will be replaced by subexpression (subtree) in gene library. This means these subexpressions (encoded into subtrees) present in the mutated antibodies will be computed. Caching these subexpressions in gene library will provide an opportunity to improve the runtime efficiency of iGEP [15]. The implementation makes use of an array, which will store cached evaluations of the subtrees. The subtree (looked as a terminal) in gene library will add a pointer to the array. When a subexpression (subtree) presented in the cache is evaluated, the cached evaluations will be used instead of re-evaluating the subexpression.

4 Experiments and Discussion

In order to examine the performance of iGEP, firstly three artificially structured symbolic regression problems were employed as test problem and compared with GEP. Later the algorithm was tested against four benchmark problems of the UCI repository [16].

4.1 Simple Symbolic Regression Problems

The simple symbolic regression problems (named T_1, T_2, and T_3) used for the experiment are respectively shown as the following:

$$f(x)=x^4+x^3+x^2+x \tag{8}$$

$$f(x)=3x^2+2x-1 \tag{9}$$

$$f(x)=4x^4+3x^3+2x^2+x \tag{10}$$

The general experiment settings are summarized as follows: the population size is 50; the function set is selected as {+, -, *, /}; the terminal condition is that the maximum number of generation is met or the precise of training is more than 0.99. Note that, for problem T_1 and T_2, the maximum of generation is set to 50, but for problem T_3, the maximum of generation is set to 5000.

The remaining parameters of GEP algorithm are set as in Table 1, and iGEP are in Table 2. Each experiment has been run for twenty times and the results were averaged.

Two algorithms can obtain the optimal solution of problem T_1, but GEP can find the optimal solution to T_2 and T_3 18 times in 20 and iGEP can find the solution every run. The average times of evaluation of the two algorithms is given in Table 3. It can be seen that the times of evaluation of iGEP are less than GEP, which shows iGEP outperforms GEP in search ability.

Table 1. Parameters of GEP algorithm

Parameter	Value
Length of head	6
Number of genes	3
Linking function	+
Selection	Roulette selection
One point crossover	0.3
two point crossover	0.3
Mutate probability	0.1

Table 2. Parameters of iGEP algorithm

Parameter	Value
Length of head	18
Size of gene library	30
Selected best antibodies	5
Editing probability (p_e)	0.4
Clone probability (p_c)	0.4
Refresh probability (p_r)	0.2
Clone factor (β)	10
Maximum Mutate probability	0.1

Table 3. Average times of evaluation of GEP vs. iGEP

	T_1	T_2	T_3
GEP	1525	1734	63916
iGEP	828	839	31328

For problem T_1, the best fitness value among all of the individuals was recorded at every generation for all of the ten runs, and the average was plotted against the generation in Fig 5 for both GEP and iGEP. The curves show that generally in the very early generations, both evolutionary processes experience a quick convergence in their best fitness. However, iGEP seems to have comparably much higher fitness than GEP for the same generation. This indicates that the iGEP outperforms in learning ability.

4.2 Classification Tests

The algorithm presented so far was tested against four benchmark problems of the UCI repository [16], which are Iris, Zoo, Win, and abalone.

The experiment settings are summarized as follows: the head length is 30; the population size is 200; the maximum of generation is set to 2000; the function set is selected as {+, -, *, /}; the terminal condition is that the maximum number of generation is met. The rest parameters are as same as in Table 1. As same as [10] we have

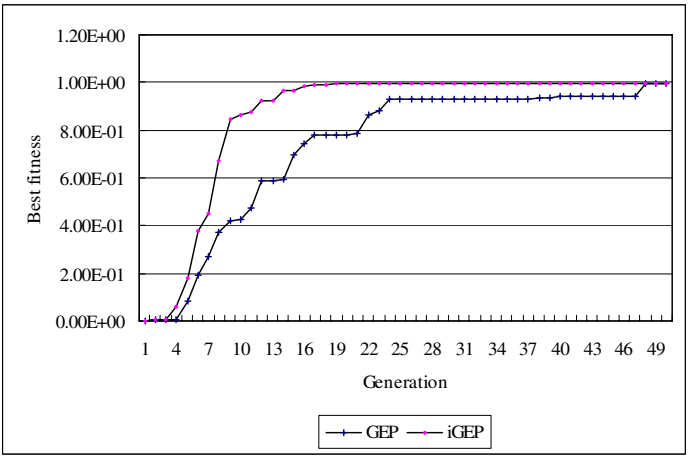


Fig. 5. The fitness convergence curves of GEP vs. iGEP for the problem T_1

also applied the five-fold cross validation method for the testing of iris, zoo and wine, and for abalone, the data are separated into two parts: the first 3133 cases as the training set and the remaining 1044 as the testing set. Finally, each experiment has been repeated for ten independent runs and the results were averaged.

Table 4 shows the optimization values. It can be observed that the behavior of the proposed algorithm was considerably good as opposed to GEP algorithm presented in ref. [10].

Table 4. Summary of the test results for iGEP vs.GEP

Dataset	Average		Best	
	iGEP	GEP	iGEP	GEP
Iris	95.8±1.0	93.7±1.4	96.9±3.2	95.3±5.7
Zoo	93.9±0.8	93.0±0.6	95.6±5.6	94.0±5.7
Wine	92.8±2.1	89.8±2.8	95.9±4.0	93.3±3.7
Abalone	57.0±3.7	50.1±4.1	59.7±3.0	53.6

4.3 Discussion

The experimental results have preliminarily validated the performance of iGEP. Since coupling this clonal selection algorithm and gene library with GEP, iGEP improve the evolvability and learning ability. Additionally, to some extent control mechanism to create an appropriate balance between exploitation and exploration is introduced into iGEP.

First, antibodies are defined as GEP expression, in order to take advantage of the flexibility and power of exploring the entire search space. Second, the best subexpression of GEP expression is selected as the solution of the problem. This is different from the standard GEP where the fitness of the individuals is give by its entire K-expression. Additionally, Subexpression gene library is adopted in iGEP in order to enhance the evolvability and the modularity learning ability and somewhat cover a coarse-grain antigen space. What's more, the hypermutation to mutate some cloned individual using some segments of gene library was introduced in order to achieve better exploitation and exploitation of the space.

5 Conclusions

GEP has been used to solve a huge variety of problems; however, in the term of evolvability and learning capability, GEP is far from perfect. Also taking inspiration from human immune system, in this paper we consider the approach to improve the evolvability of GEP based on Artificial Immune Systems (AIS) and subexpression learning. iGEP is expected to have the following features: the best subexpression of GEP expression is selected as the solution of the problem, and some segments of gene library are used for hypermutation and receptor editing. The algorithm is tested on some benchmark problems of the UCI repository, and in terms of convergence rate and computational efficiency the results show that iGEP outperforms the standard GEP. Future research on iGEP will primarily focus on analyzing the emergency of subexpression in iGEP and enhancing the learning capability in view of evolvability.

Acknowledgments. This work is supported by the Research Foundation for Outstanding Young Teachers of China University of Geosciences (CUGQNL0644).

References

1. Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, 2nd edn. Springer, Germany (2006)
2. Chi, Z.: Gene Expression Programming and Rule Induction for Domain Knowledge Discovery and Management, Doctoral dissertation, Department of Computer Science, University of Illinois at Chicago, Chicago (2003)
3. Lopes, H.S., Weinert, W.R.: EGIPSYS: an Enhanced Gene Expression Programming Approach for Symbolic Regression Poblems. International Journal of Applied Mathematics and Computer Science 14(3), 375–384 (2004)
4. De Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transaction on evolutionary computation, special issue on artificial immune system, 239–251 (2002)
5. Cayzer, S., Smith, J.: Gene Libraries: Coverage, Efficiency and Diversity. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 136–149. Springer, Heidelberg (2006)
6. Kim, J., Bentley, P.J.: Immune Memory and Gene Library Evolution in the Dynamic Clonal Selection Algorithm. Genetic Programming and Evolvable Machines 5, 361–391 (2004)
7. Zhuo, K., Kang, L., Li, Y.: A New Automatic Programming Method for Program Reuse. In: Proceedings of the 3rd International Conference on Neural, Parallel and Scientific Computations, Atlanta, USA (2006)
8. Karakasis, V.K., Stafylopatis, A.: Data Mining based on Gene Expression Programming and Clonal Selection. In: CEC 2006, pp. 514–521 (2006)
9. Xin, L., Zhou, C., Xiao, W., Nelson, P.C.: Direct Evolution of Hierarchical Solutions with Self-emergent Substructures. In: ICMLA 2005, pp. 337–342 (2005)
10. Xin, L., Zhou, C., Xiao, W., Nelson, P.C.: Prefix Gene Expression Programming. In: GECCO 2005 (2005)
11. Valiant, L.G.: Evolvability. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708, pp. 22–43. Springer, Heidelberg (2007)
12. Lones, M.A., Tyrrell, A.M.: Enzyme Genetic Programming. In: Amos, M. (ed.) Cellular Computing. Genomics and Bioinformatics Series, pp. 18–41. Oxford University Press, Oxford (2003)
13. Mihai, O., Grosan, C.: A Comparison of Several Linear Genetic Programming Techniques. Complex-Systems 14(4), 285–313 (2003)
14. Mihai, O.: Multi Expression Programming, Technical Report, Babes-Bolyai Univ, Romania (2006)
15. Keijzer, M.: Alternatives in Subtree Caching for Genetic Programming. In: Keijzer, M., O'Reilly, U.-M., Lucas, S., Costa, E., Soule, T. (eds.) EuroGP 2004. LNCS, vol. 3003, pp. 328–337. Springer, Heidelberg (2004)
16. Blake, L.C., Merz, J.C.: UCI Repository of Machine Learning Databases. University of California, Department of Information and Computer Science, Irvine (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Correlation between Mutations and Self-adaptation in Evolutionary Programming

Yong Liu

The University of Aizu
Aizu-Wakamatsu, Fukushima 965-8580, Japan
yliu@u-aizu.ac.jp

Abstract. It has been taken for granted that long jumps of Cauchy mutation in a fast evolutionary programming (FEP) increase the probability of finding a near-optimum when the distance between the current search point and the optimum is large, but decrease the probability when such distance is small [1]. By explicitly measuring the search step sizes, this paper gives sound evidence that not long jumps but large variances in Cauchy mutation have contributed to the better performance of FEP than that of classical evolutionary programming (CEP). It has been discovered that smaller step-size mutations among Cauchy mutations had led to the faster convergence of FEP in some test functions, while these helpful Cauchy mutations could actually have shorter search step sizes than Gaussian mutations used in CEP. The reason that Cauchy mutations could have shorter step sizes than Gaussian mutations is that Cauchy mutations and Gaussian mutations could radically alter self-adaptation in FEP and CEP. This paper further discusses the correlation between mutations and self-adaptation in CEP and FEP.

1 Introduction

Evolutionary programming (EP) [2,3] is a population-based generate-and-test algorithms that use mutations only to generate new solutions. The generating process of an offspring mutated from a parent is controlled completely by the search bias of a mutation that consists of its step size and search directions.

The step size is crucial for the performance of a search operator [1,4]. Different search operators often have different step sizes, and thus are appropriate for different problems as well as different evolutionary search stages for a single problem. Since the global optimum is generally unknown in real world applications, it is impossible to know a priori what step sizes should be applied in an EP. Self-adaptation was therefore adopted in EP with Gaussian mutation in which strategy parameters are encoded and evolved together with each individual [3]. Through evolving strategy parameters, the step sizes could therefore be controlled in EP.

Since a step size is actually a product of the strategy parameter and the absolute value of the random value generated from its random distribution in one dimensional case, the step size is also controlled by the distributions of random

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 58–66, 2008.
© Springer-Verlag Berlin Heidelberg 2008

numbers used in the mutations. Different random distributions lead to the different step sizes in mutations. For example, for a Gaussian mutation using Gaussian random number with standard deviation 1 and a Cauchy mutation applying Cauchy random number with the scale parameter 1, the Cauchy mutations tend to have larger average mutation steps than those of Gaussian mutations when the strategy parameter is the same. The relationship between the search step size and the probability of finding a global optimum had been explored, and a fast EP using Cauchy mutations had been proposed in [1].

It is important to design more efficient EP. However, it might be more important to understand why and how mutations have worked. It had been stated in [1] that long jumps of Cauchy mutation in a fast evolutionary programming (FEP) increase the probability of finding a near-optimum when the distance between the current search point and the optimum is large, but decrease the probability when such distance is small. However, no explicit measurement of Cauchy mutation step sizes had been given in [1]. When the mutation step sizes depend also on the strategy parameter, it would be unclear how much Cauchy mutation step sizes could be longer than Gaussian mutation step sizes if the strategy parameters might be changed differently in FEP and CEP. If the strategy parameters would be much smaller in FEP, step sizes of Cauchy mutations in FEP might be shorter than than those of Gaussian mutations in CEP. Even for the same strategy parameter, some mutations among Cauchy mutations could have long jumps, while others could have relatively short jumps. When not all Cauchy mutations have made contributions to the evolution, it is also unclear whether long jumps or short jumps among all Cauchy mutations have actually led to the evolution. This paper firstly addresses these two unsolved issues by observing how mutation step sizes change in the evolution.

Another issue to be dealt with in this paper is how different mutations and self-adaptation would affect each other. It was assumed in [1] that different mutations in FEP and EP would not have altered self-adaptation much when no any explicit change has been made in self-adaptation. However, such assumption turns out not to be true. This paper shows how significantly they two would affect each other by examining the changes of strategy parameter values.

The rest of this paper is organized as follows. Section 2 describes the major steps of CEP and FEP discussed in this paper. Section 3 presents the experimental results and discussions on mutation step sizes and correlation between different mutations and self-adaptation. Finally, Section 4 concludes with some remarks and future research directions.

2 Function Optimization by Evolutionary Programming

A global minimization problem can be formalized as a pair (S, f), where $S \subseteq R^n$ is a bounded set on R^n and $f : S \mapsto R$ is an n-dimensional real-valued function. The problem is to find a point $\mathbf{x}_{min} \in S$ such that $f(\mathbf{x}_{min})$ is a global minimum on S. More specifically, it is required to find an $\mathbf{x}_{min} \in S$ such that

$$\forall \mathbf{x} \in S : f(\mathbf{x}_{min}) \leq f(\mathbf{x}),$$

where f does not need to be continuous but it must be bounded. This paper only considers unconstrained function optimization.

According to the description by Bäck and Schwefel [3,1], CEP for function optimization is implemented as follows:

1. Generate the initial population of μ individuals, and set $k = 1$. Each individual is taken as a pair of real-valued vectors, $(\mathbf{x}_i, \eta_i)$, $\forall i \in \{1, \cdots, \mu\}$, where $\mathbf{x}_i$'s are objective variables and η_i's are standard deviations for Gaussian mutations (also known as strategy parameters in self-adaptive evolutionary algorithms).
2. Evaluate the fitness score for each individual $(\mathbf{x}_i, \eta_i)$, $\forall i \in \{1, \cdots, \mu\}$, of the population based on the objective function, $f(\mathbf{x}_i)$.
3. Each parent $(\mathbf{x}_i, \eta_i)$, $i = 1, \cdots, \mu$, creates a single offspring $(\mathbf{x}_i', \eta_i')$ by: for $j = 1, \cdots, n$,

$$\eta_i'(j) = \eta_i(j) \exp(\tau' N(0,1) + \tau N_j(0,1)) \quad (1)$$

$$x_i'(j) = x_i(j) + \eta_i(j) N_j(0,1), \quad (2)$$

   where $x_i(j)$, $x_i'(j)$, $\eta_i(j)$ and $\eta_i'(j)$ denote the j-th component of the vectors $\mathbf{x}_i$, $\mathbf{x}_i'$, η_i and η_i', respectively. $N(0,1)$ denotes a normally distributed one-dimensional random number with mean 0 and standard deviation 1. $N_j(0,1)$ indicates that the random number is generated anew for each value of j. The factors τ and τ' are commonly set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$ [3].
4. Calculate the fitness of each offspring $(\mathbf{x}_i', \eta_i')$, $\forall i \in \{1, \cdots, \mu\}$.
5. Conduct pairwise comparison over the union of parents $(\mathbf{x}_i, \eta_i)$ and offspring $(\mathbf{x}_i', \eta_i')$, $\forall i \in \{1, \cdots, \mu\}$. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is no larger than the opponent's, it receives a "win."
6. Select the μ individuals out of $(\mathbf{x}_i, \eta_i)$ and $(\mathbf{x}_i', \eta_i')$, $\forall i \in \{1, \cdots, \mu\}$, that have the most wins to be parents of the next generation.
7. Stop if the halting criterion is satisfied; otherwise, $k = k+1$ and go to Step 3.

FEP discussed in this paper is exactly the same as the CEP described in except for Eq.(2) which is replaced by the following [1]:

$$x_i'(j) = x_i(j) + \eta_i(j)\delta_j, \quad (3)$$

where δ_j is a Cauchy random variable with the scale parameter $t = 1$, and is generated anew for each value of j. It is worth indicating that we leave Eq.(1) unchanged in FEP in order to keep our modification of CEP to a minimum. It is also easy to investigate the impact of the Cauchy mutation on EP when other parameters are kept the same.

3 Experimental Studies

3.1 Benchmark Functions and Experimental Setup

Twenty-three benchmark functions used in [2,3,1] were tested in our experimental studies. Among the twenty-three benchmark functions, there are unimodal

Table 1. The six benchmark functions used in our experimental study, where n is the dimension of the function, f_{min} is the minimum value of the function, and $S \subseteq R^n$

Test function	n	S	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]^n$	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$[-100, 100]^n$	0
$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10)]$	30	$[-5.12, 5.12]^n$	0
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^n$	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	0

functions f_1 to f_7, multimodal functions f_8 to f_{13} with many local minima, and low-dimensional multimodal functions f_{14} to f_{23} with a few local minima. Because of page limits, only the results of the six functions among the twenty-three functions listed in Table 1 had been included in this paper.

In all experiments, the same self-adaptive method (i.e., Eq.(1)), the same population size $\mu = 100$, the same tournament size $q = 10$ for selection, the same initial $\eta = 3.0$, the same lower bound 0.001 for η, and the same in initial population were used for both CEP and FEP. The initial population was generated uniformly at random in the specified ranges in each test function. as specified in Table 1.

3.2 Performance Comparison

The results of mean of best solutions of 50 independent runs of CEP and FEP at generation 2000 are summarized in Table 2. For unimodal functions, CEP outperformed FEP on f_1 and f_2, while FEP achieved better results on f_3. FEP had displayed much better results than CEP on multimodal functions f_9, f_{10}, and f_{11}.

Figure 1 apparently shows that that FEP performs better than CEP in terms of convergence rate although CEP's final results were better than FEP's for unimodal functions f_1 and f_2. Function f_1 is the simple sphere model studied by many researchers. In the beginning of evolution in f_1, FEP dispays a faster convergence rate than CEP due to its better global search ability. It quickly approaches the neighborhood of the global optimum and reaches approximately 0.001 in around 1200 generations, while CEP can only reach approximately 0.01.

Table 2. Comparison between CEP and FEP on the results of the mean best function values found at generation 2000 over 50 runs

	f_1	f_2	f_3	f_9	f_{10}	f_{11}
CEF	2.9×10^{-5}	0.022	244.01	85.79	7.758	0.117
FEP	2.8×10^{-4}	0.067	129.2	1.76	0.013	0.021

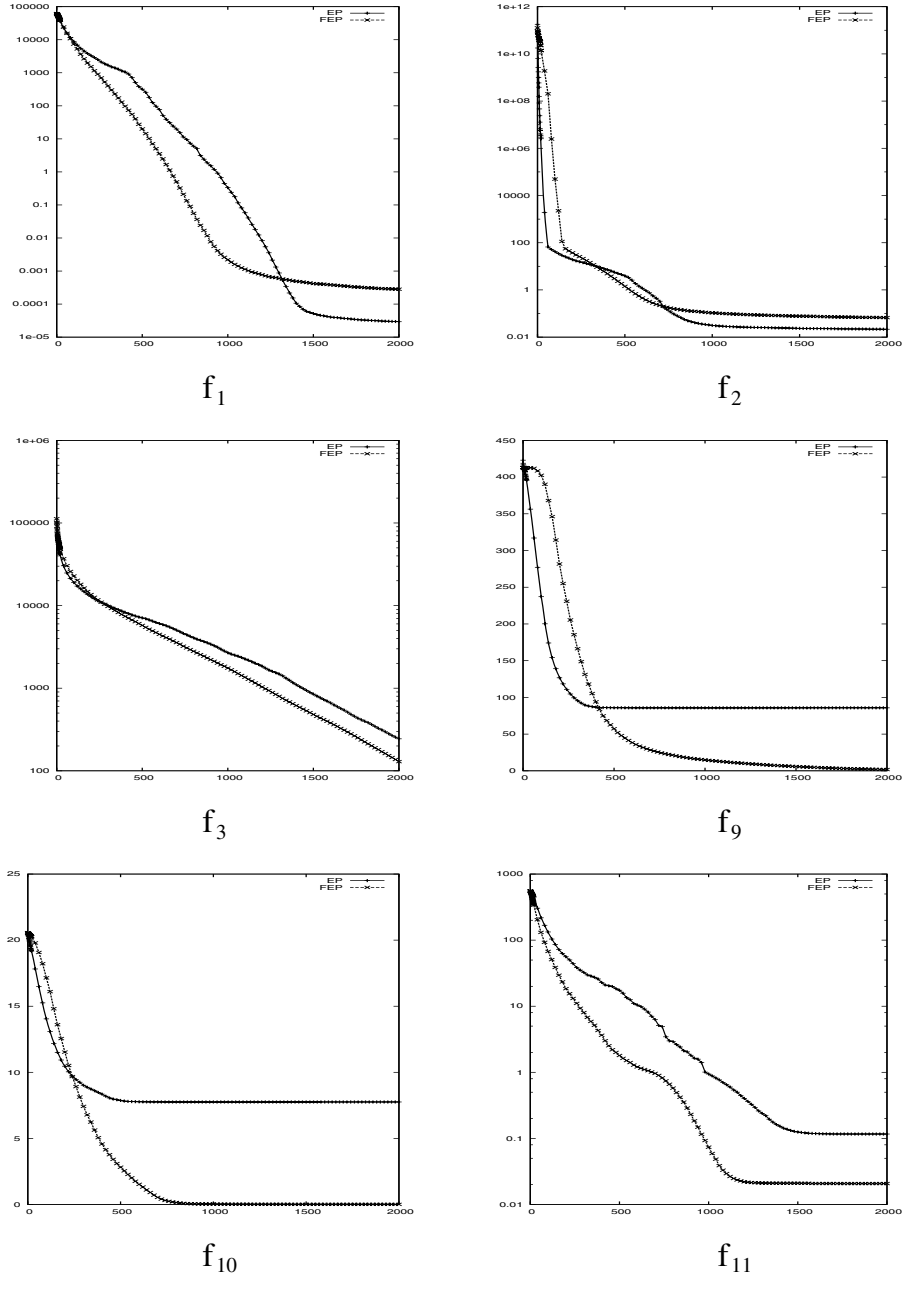


Fig. 1. Comparison between EP and FEP on the evolution process minimizing f_1, f_2, f_3, f_9, f_{10}, and f_{11}. The vertical axis is the mean of best solutions of 50 independent runs, and the horizontal axis is the number of generations.

After that, FEP's convergence rate reduces substantially, while CEP maintains a nearly constant convergence rate throughout the evolution. Finally, CEP overtakes FEP at around 1350th generation. For multimodal functions, although CEP improved faster at early evolution for f_9 and f_{10}, CEP appears to become trapped in poor local optima and unable to escape from them since then. In contrast, FEP performs significantly better than CEP consistently for these multimodal functions.

3.3 Correlation between Mutations and Self-adaptation

The results given in Section 3.2 had been reported in [1]. The reason given for the better performance of FEP in [1] is that long jumps of Cauchy mutation in a fast evolutionary programming (FEP) increase the probability of finding a near-optimum when the distance between the current search point and the optimum is large. By explicitly measuring the search step sizes, this Section gives sound evidences that not long jumps but large variances in Cauchy mutation have contributed to the better performance of FEP.

Let SS, WS, and FS denote the average of step sizes over all mutations, winning mutations, and failing mutations at each generation, respectively. Winning mutations include those mutations that generate offspring having smaller function values than their parents, while failing mutations consist of the rest mutations that generate offspring having no less function values than their parents. Figure 2 shows the evolution process of the average step sizes of all mutations, winning mutations, and failing mutations from 50 runs for CEP and FEP. It is obvious that the average step sizes of failing mutations are close to the average of all mutations for both Gaussian and Cauchy mutations. In contrast, the average step sizes of winning mutations are much smaller. Although the average step sizes FEP_SS of Cauchy mutations are generally larger than CEP_SS through the evolution in optimizing all six functions, it is surprised that the average step sizes FEP_WS of winning Cauchy mutations could even be smaller than EP_WS of winning Gaussian mutation in functions f_1, f_3, and f_{11}. Such a fact looks contradict to the analytical result that FEP's long jumps increase the probability of finding a near-optimum when the distance between the current search point and the optimum is large, but decrease the probability when such distance is small. It should be borne in mind that such an analytical result is obtained in minimization of one-dimensional functions. It might not be applied to multi-dimensional functions when changes in different dimensions could easily affect and conflict each others in function values. There is no such a conflict in optimizing one-dimensional functions. Actually, the improving probability of FEP's long jumps is much smaller than that of CEP's short jumps.

It should be noted that only the winning mutations are helpful at most time of evolution. Although the offspring from failing mutations could possibly survive in EP with random competition selection, only offspring from winning mutations could be selected to the next generation in evolution strategies [3]. While the averge winning step sizes FEP_WS of Cauchy mutations are smaller than

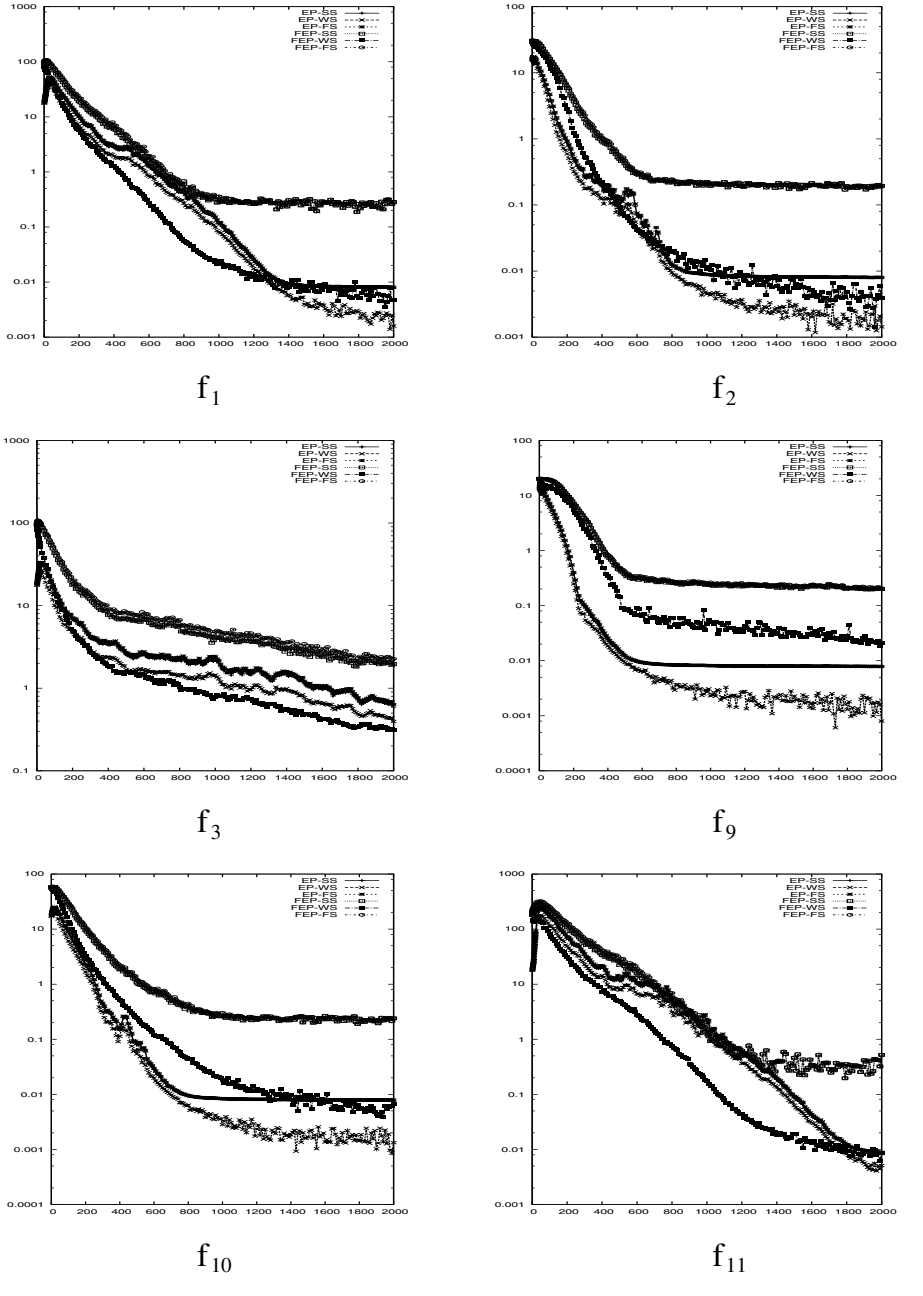


Fig. 2. Comparison between EP and FEP on the average step sizes of all mutations, winning mutations, and failing mutations from 50 runs for f_1, f_2, f_3, f_9, f_{10}, and f_{11}. The vertical axis is the step sizes, and the horizontal axis is the number of generations.

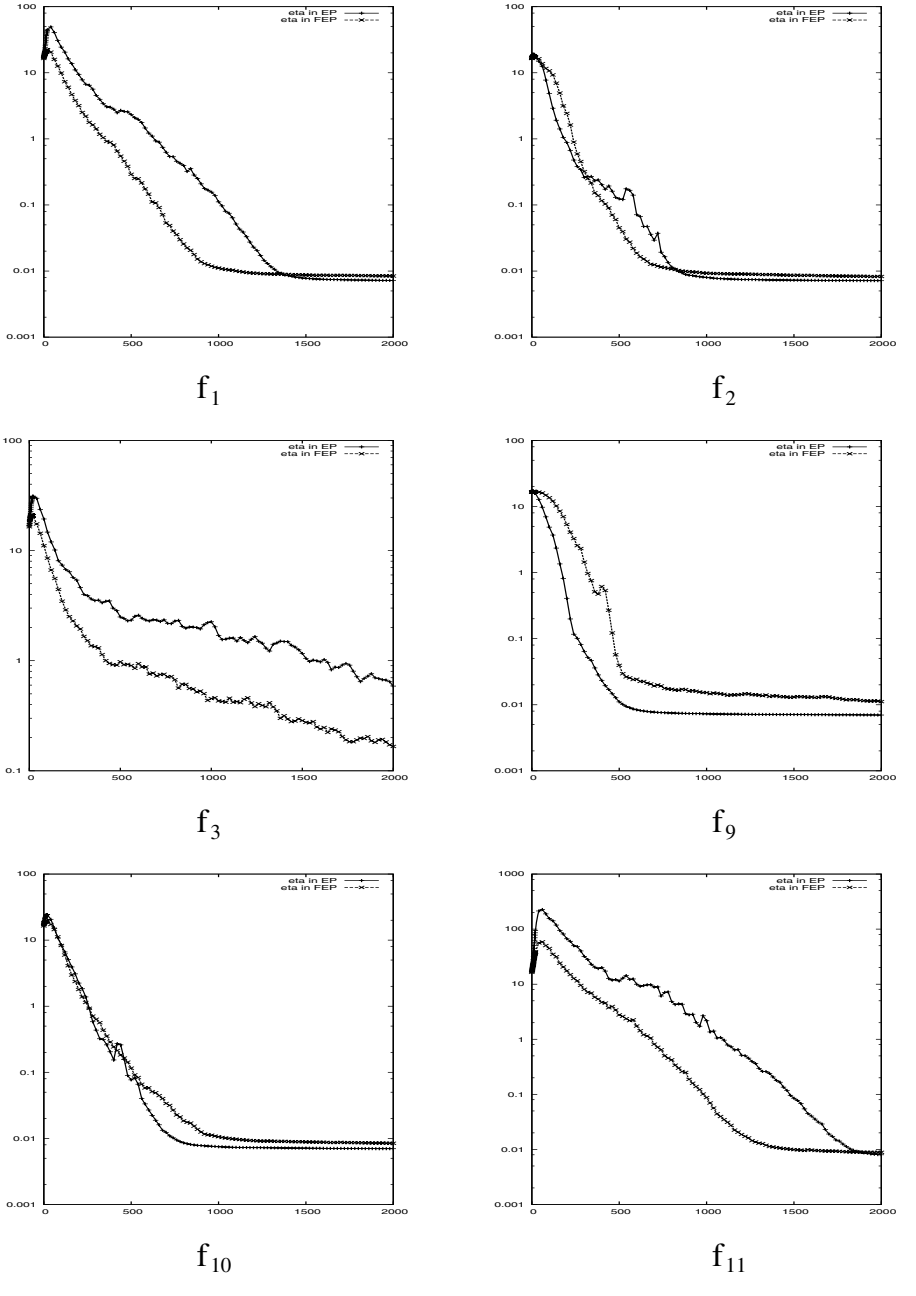


Fig. 3. Comparison between EP and FEP on the average of the values of η in self-adaptation from 50 runs for f_1, f_2, f_3, f_9, f_{10}, and f_{11}. The vertical axis is the values of η, and the horizontal axis is the number of generations.

those of Gaussian mutations in minimizing f_1, f_3, and f_{11}, the winning Cauchy mutations have larger step sizes for the rest three functions. Such a difference among FEP_WS is caused by the different changes in self-adaptation for different mutations and different functions as well. Figure 3 displays the evolution process of the average of the values of η in self-adaptation from 50 for CEP and FEP. It is clear that η in FEP is smaller than its value in CEP for f_3 and f_{11}, and larger than its value in CEP for f_9. For other three functions, the values of η in CEP and FEP are mixed. Generally speaking, the winning mutation step sizes are smaller when η is smaller. The η and the winning mutation step sizes WS follow the same trend in Figures 2 and 3.

4 Conclusions

The purpose of this paper is not to show whether FEP could perform better than CEP, but discover why FEP had achieved the better results than CEP. Not the long jumps but large variances in Cauchy mutations have made FEP more powerful although those successful Cauchy mutations could have shorter jumps at average.

This paper also made another discovery of correlation between mutations and self-adaptation in CEP and FEP. Compared to the self-adaptation of η in CEP, η tends to decrease faster in FEP for unimodal functions, while it might decrease slower in FEP for multimodal functions. Therefore, the winning Cauchy mutations could have relatively large or small step sizes on minimizing different functions. It is possible that the average step sizes of successful Cauchy mutations could be much smaller.

References

1. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 3(2), 82–102 (1999)
2. Fogel, D.B.: System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling. Ginn Press, Needham Heights (1991)
3. Bäck, T., Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation 1(1), 1–23 (1993)
4. Lee, C.Y., Yao, X.: Evolutionary programming using the mutations based on on the Lévy probability distribution. IEEE Transactions on Evolutionary Computation 8(1), 1–13 (2004)

Representations of Evolutionary Electronics

Xuesong Yan, Pan Fang, Qingzhong Liang, and Chenyu Hu

School of Computer Science, China University of Geosciences, Wu-Han, 430074, China
Research Center for Space Science and Technology, China University of Geosciences, Wu-Han, 430074, China
yanxs1999@hotmail.com

Abstract. For the evolutionary algorithm, the representation of the electronic circuit has two methods, one kind is code with the electronic circuit solution space, the other is code with the problem space. How weighs one representation quality may think the following questions? The first is the code method should as far as possible complete, it is say for the significance solution circuit or the optimize solution obtains in the problem space may represented by this code method. The second is the code method should speeds up the convergence speed of the algorithm search. The hardware representation methods mainly include binary bit string representation, tree representation, Cartesian Genetic Programming representation and other representations. In this paper, we will introduce the representations of the binary bit string and Cartesian Genetic Programming in detail, then give some examples of the two representations.

1 Introduction

Evolutionary Electronics[1-5] applies the concepts of genetic algorithms to the evolution of electronic circuits. The main idea behind this research field is that each possible electronic circuit can be represented as an individual or a chromosome of an evolutionary process, which performs standard genetic operations over the circuits. Due to the broad scope of the area, researchers have been focusing on different problems, such as Field Programmable Gate Array (FPGA) mapping [6], optimization of combinational and sequential digital circuits [7], synthesis of digital circuits [8], synthesis of operational amplifiers, transistor size optimization and so on.

Of great relevance are the works focusing on "intrinsic" hardware evolution in which fitness evaluation is performed in silicon, allowing a higher degree of exploration of the physical properties of the medium. This particular area is frequently called Evolvable Hardware, though this terminology may also be used in a broader sense. Now the research of the evolvable hardware lies in the hardware representation and the concrete example, not the algorithm, so the hardware representation and code is the most important research contents. No matter is carries on the electronic circuit design and optimize using the evolvable hardware, or carries on the evolvable hardware research, the basic problems are the domain knowledge and hardware representation. For the evolutionary algorithm, the representation of the electronic circuit has two methods,

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 67–78, 2008.
© Springer-Verlag Berlin Heidelberg 2008

one kind is code with the electronic circuit solution space, the other is code with the problem space. How weighs one representation quality may think the following questions? The first is the code method should as far as possible complete, it is say for the significance solution circuit or the optimize solution obtains in the problem space may represented by this code method. The second is the code method should speeds up the convergence speed of the algorithm search.

The hardware representation methods mainly include binary bit string representation, tree representation, Cartesian Genetic Programming representation and other representations. In this paper, we will introduce the representations of the binary bit string and Cartesian Genetic Programming in detail, then give some examples of the two representations.

2 Binary Bit String Representation

2.1 Concept of Binary Bit String Representation

Programming Logic Device (PLDs) are hardware devices of which architecture can be determined by downloading a binary bit string, called architecture bits. Architecture bits are the compilation result from higher level of hardware description such as boolean functions and truth tables. On the other hand, GAs are robust search algorithms where multiple chromosomes (usually represented as a binary bit string) are used to find better solutions.

The basic idea of EHW [9]is to regard the architecture bits of PLDs as chromosomes of GAs and to find out better hardware structure by GAs, as shown in Figure 1. Once a good chromosome is found by GAs, it is downloaded into PLDs on the spot. A PLD consists of logic cells and a fuse array. A logic cell can realize some logic functions and one of the functions is selected by specifying a bit string. In other words, the logic cell's function is programmable.

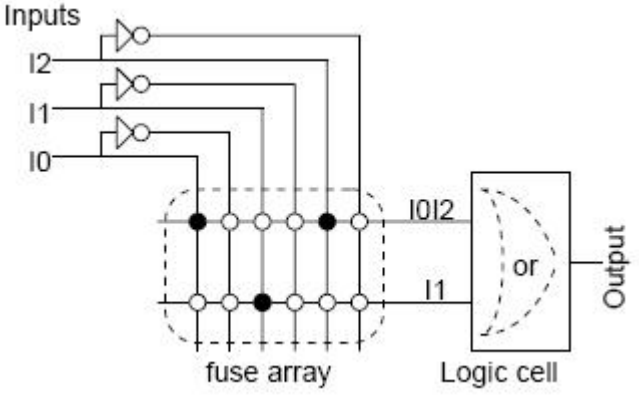


Fig. 1. A simplified PLD(Programmable Logic Device) structure

Fuse array determines the interconnection between external inputs and logic cells. It actually generates AND-term signals by specifying the switches on the fuse array. For example, in Figure 2, the black dots on the fuse array indicate that external inputs are connected with a line of the fuse array. The switch setting on the first row of Figure 2 indicates that I0 and I2 are entered into the first row, generating a AND-term, I0, I2. Similarly, I1 is generated on the second row. These AND-terms then enters into the logic cell of which function is chosen to be an OR gate. Notice that such switch setting are also regarded as a binary bit string. Thus, both of the fuse array and logic cell functions are controlled by a binary string. The key idea of EHW is to regard such a binary bit string as chromosome of GAs and find out the best hardware structure by GAs.

There are some research work on EHW was directly focused on the architecture bits[9]; bits for fuse array and bits for logic cells are regarded as genotype. Although this genotype representation is straightforward, the researchers succeeded in hardware evolution of both combinatorial logic circuits and sequential logic. However, this genotype representation has inherent limitations, since the fuse array bits are fully included in the genotype even in the case that only a few bits are effective. This causes the increase of the chromosome length, increasing the GA execution time.

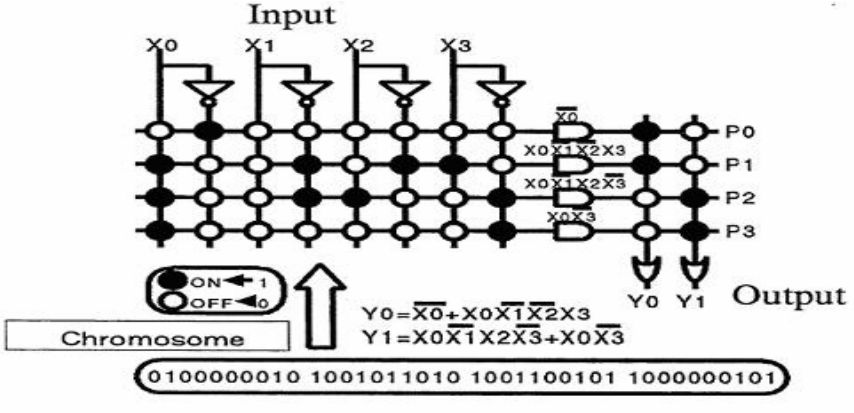


Fig. 2. Chromosome of PLD

2.2 Algorithm Based on Binary Bit String Representation

2.2.1 Modeling for Circuit Representation

For this representation we design the algorithm, in our algorithm, the AND-gate array with M column and K row, the OR-gate array with N column and K row, then in the model have two array with M×K and N×K, the length of the chromosome will be calculate by this equation:

$$\text{Length} = (2\times M+N)\times K \tag{1}$$

In here, we give a example, for a circuit with four input and one output. With our model, the length of the chromosome is 36, the initial bit-string with constraints: in the

bit-string the two adjoin bits will not be one in the same time. For the following individual, the red denotes the OR-gate.

Individual: 010010101101010000001001010010001101

For this individual we can know the logic function is: F=ACD+A$\overline{C}$ $\overline{D}$ and the circuit struction is like fig..

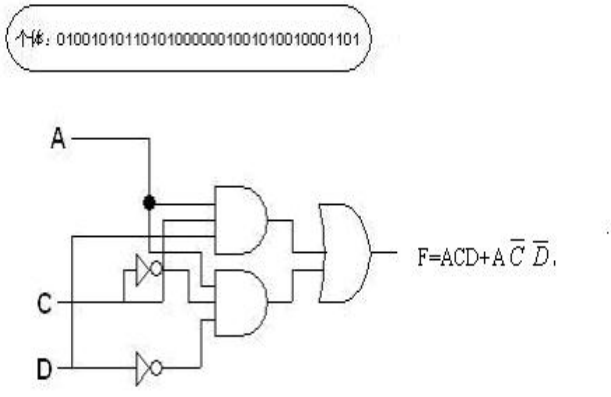


Fig. 3. Circuit and Logic Expression

2.2.2 Individual Evaluate and Operator Design

In the algorithm, the most important is the fittness function, because only the fittness function can define the individual good or bad, so in our algorithm, we describe the fittness function like this: we can get the logic function from the bit-string and detect the function maybe fit the truth table, then calculate the gates number of the corresponding logic circuit, the gates number more little and the fittness value more best. The whole process is like this: According the chromosome bit-string, we can get a logic function expression, we can put the values in the truth table into the expression and calculate a value. If the value is equal to the output then we know the individual is the right individual. If the input variant have four: A、B、C、D, then we can statistic the number of using the four variants A、B、C、D and the number of using the logic gates. In the evolution process, we must maintain the individual, which has fewer variants, logic gates and can Satisfies the function.

In our algorithm, we deign the genetic operator according the circuit function. The crossover method we use the two-position crossover and the mutation we use one point mutation. The length of the chromosome is 36 bit, the process of crossover operation like following:

Individual 1 : 100001**010101010011**100010101010000001

Individual 2 : 001010**000010001101**011001001100110000

In the crossover, we select two number: m and n in random, in here m<36 n<36.

If m<n, we exchange the string from n to m in individual 1 and individual 2, then generate the two new individuals.

New individual 1 : 10000100001000111001010101000001
New individual 2 : 00101001010101001101100100110000

The mutation operation like the crossover, the step is following:

individual : 10000101010101001110001010101000001

generate the number m in random, in here m<36.

For this operation, we will find the value a according the point m and do the operation 1-a, the new individual is like this:

New individual : 10000101000101001110001010101000001

2.2.3 Algorithm Framework

```
begin
{
Generate the population of the individual PopSize P(0),
t=0;
Calculate the fitness values of the individuals;
   while(< Max-generation) Do
     {
         do the genetic operation with the individual;
          according the fitness value and selection
stratigy calculate the selection probability P_i and the
crossover probability P_c of the individuals;
       do the crossover operation according to the P_i and
calculate the child individual fitness value, if the
fitness value is better then the child population replace
the parent individual and update the population;
      do the mutation operation according to the P_i to
select N1(N1<Popsize)individual, and calculate the fitness
value, if the fitness value is better then the child
population replace the parent individual and update the
population;
generate the new population P(t+1);
t=t+1 ;
     }
}
```

2.2.4 Experiment Result

For the model and the operator, we test our algorithm use the following example. In here the PLD model is four input and one output, and the test function is ture-table, the input is: A, B, C, D and the output is F.

Table 1. Ture-table of the problem

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

According to the ture-table, we use our algorithm design the circuit, the parameters for the algorithm are as Table 2.

Table 2. The parameters for the algorithm

Number of generations	1000
Population Size	100
Mutation rate	0.003
Crossover rate	0.08
Selection range	100

In the table is the experiment results:

From the experiment results, we can know use the algorithm design circuit can get different design scheme, and can verify our algorithm is effective for the problem of circuit design.

Table 3. Experiment results

Individual string	Number of the result	Number of Gate	Number of Variant
10001000010100000110101000110001000 1	16	4	3
10001010110100000010001000110100000 0	16	4	4
10100100110001000110101000010101000 0	16	5	4
10100110110001000110001000110100000 1	16	6	5
01010010010001000110100010110101010 0	15	21	4
10100001110001010100101010010011001 1	15	21	6
10000000110101010110101010110101010 1	14	21	4
10000000110101010110101010110010101 1	14	21	7

3 Cartesian Genetic Programming Representation

3.1 Concept of Cartesian Genetic Programming Representation

In our algorithm, the chromosome representation we use Cartesian Genetic Programming, which is proposed by Miller[10]. CGP is Cartesian in the sense that the method considers a grid of nodes that are addressed in a Cartesian coordinate system. In the CGP the genotype is represented as a list of integers that are mapped to directed graphs rather than trees. One motivation for this is that it uses graphs which are more general than trees. In CGP the genotypes are of fixed length but the phenotypes are of variable length according to the number of unexpressed genes. This representation is based on the FPGA of Xilinx Virtex-II. The starting point in this technique is to consider, for each potential design, a geometry (of a fixed size array) of uncommitted logic cells that exist between a set of desired inputs and outputs. The uncommitted logic cells are typical of the resource provided on the Xilinx FPGA part under consideration. An uncommitted logic cell refers to a two-input, single-output logic module with no fixed functionality. The functionality may then be chosen, at the implementation time, to be any two input variable logic function. In this technique, a chromosome is defined as a set of interconnections and gate level functionality for these cells from outputs back toward the inputs based upon a numbered rectangular grid of the cells themselves, as in Fig. 4. The inputs that are made available are logic '0', logic '1', all primary inputs and primary inputs inverted. To illustrate this consider a 3 x 3 array of logic cells between two required primary inputs and two required outputs.

The inputs 0 and 1 are standard within the chromosome, and represent the fixed values, logic '0' and logic '1' respectively. The inputs (two in this case) are numbered 2 and 3, with 2 being the most significant. The lines 4 and 5 represent the inverted inputs 2 and 3 respectively. The logic cells which form the array are numbered column-wise from 6 to 14. The outputs are numbered 13 and 11, meaning that the most significant

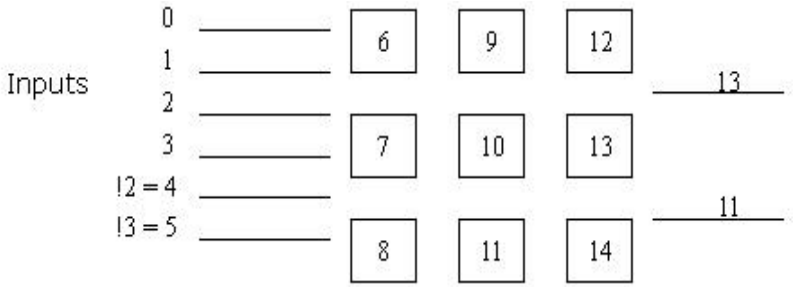


Fig. 4. A 3 x 3 geometry of uncommitted logic cells with inputs, outputs and netlist numbering

output is connected to the output of cell 13 and the least significant output is connected to the output of cell 11. These integer values, whilst denoting the physical location of each input, cell or output within the structure, now also represent connections or *routes* between the various points. In other words, this numbering system may be used to define a netlist for any combinational circuit. Thus, a chromosome is merely a list of these integer values, where the *position* on the list tells us the cell or output which is being referred to, while the value tells us the connection (of cell or input) to which that point is connected, and the cells functionality.

Each of the logic cells is capable of assuming the functionality of any two-input logic gate, or, alternatively a 2-1 *multiplexer* (MUX) with single control input. In the geometry shown in Fig. 5. a sample chromosome is shown below:

0 2 -1 1 3 -5 2 4 3 0 8 -10 7 8 -4 6 11 9 6 4 -9 2 11 7 13 11

Fig. 5. A typical netlist chromosome for the 3 x 3 geometry of Fig.1

Notice, in this arrangement that the chromosome is split up into groups of three integers. The first two values relate to the connections of the two inputs to the gate or MUX. The third value may either be positive - in which case it is taken to represent the control input of a MUX - or negative - in which case it is taken to represent a two-input gate, where the modulus of the number indicates the function according to Table 1 below. The first input to the cell is called A and the second input called B for convenience. For the logic operations, the C language symbols are used: (i) & for AND, (ii) | for OR, (iii) ^ for exclusive-OR, and (iv) ! for NOT. There are only 12 entries on this table out of a possible 16 as 4 of the combinations: (i) all zeroes, (ii) all ones, (iii) input A passed straight through, and (iv) input B passed straight through are considered trivial - because these are already included among the possible input combinations, and they do not affect subsequent network connections in cascade.

This means that the first cell with output number 6 and characterised by the triple {0, 2, -1} has it's a input connected to '0', its B input connected to input 2, and since the third value is -1, the cell is an AND gate (thus in this case always produces a logical output of 0). Picking out the cell who's output is labelled 9, which is characterised by

the triple {2, 6, 7}, it can be seen that its A input is connected to input 2 and its B input is connected to the output of cell 6, while since the third number is positive the cell is a MUX with control input connected to the output of cell 7.

3.2 Algorithm Based on Cartesian Genetic Programming

3.2.1 Genetic Operation

Mutation Operation: In this algorithm, the mutation operate is for the triple, each or some of the gene in the triple do the mutate, the mutation probability is 0.7, that is to say, each triple do the mutate with this probability and the mutation domain is 20.

Crossover Operation: Select the fittest individual being the parent for the next generation in the population. Each other individual do the crossover operate with this parent.

Fitness Measure: The test of whether the evolved circuits perform the desired logic translation of inputs to outputs is achieved by running all test inputs through the network, and compared the results with the desired functionality in a bit-wise fashion. A PLA file (truth table) contains the target function, and this is used as a basis for comparison. The percentage of total correct outputs in response to appropriate inputs is then used as the fitness measure for the genetic algorithm. In other words, the nearer the evolved circuit comes to performing with desired functionality, the fitter it is deemed to be. In our algorithm we uses this method: comparing all the outputs of a individual with the desired outputs. For a set of input in the true table, if there is one bit which belongs to the outputs of the circuit individual, different to the desired value, though deemed the circuit functionality useless for this input. When total outputs are equal to the desired outputs, then the fitness value added 1.

3.2.2 Algorithm Framework

Our algorithm through populations evolution to get the fittest solution, in the evolutionary process, we use the genetic operation to guarantee the individual diversity in the population, thus enables the population to be able rapid convergence. The electric circuit automation designs problem has it's particularity, compared with other optimized problem the different place for the electric circuit automation design lies in the ultimate objective is explicit which is to get the circuit solution, in this circuit the output of a individual fit the true table value. For this idea, we design our algorithm like following:

```
Step 1: random generation of the initial populations P10
and initial the counter t=0;

Step 2: select the individuals from populations according
to the fitness value and crossover with these individuals
to get the new individuals Q;

Step 3: for the new individuals use the mutation operator
according to probability to generate the new population Pt;
```

```
Step 4: t=t+1;

Step 5: if the terminate conditions satisfied turn to step
6, else turn to step 2;

Step 6: output the new population Pt;
```

From the framework with our algorithm, it is show this algorithm has guaranteed the population's diversity and causes the population not to trap into the local optima.

3.2.3 Experiment Results

This section shows the implementation of our experiment case two-bit full adder. The parameters for the algorithm are as Table 2.

Table 4. The parameters for the algorithm

Number of generations	2000
Population Size	30
Chromosome length	39
Mutation rate	0.7
One-point recombination rate	1.0
Selection range	100

A two-bit full adder, with a truth table with 5 inputs and 3 outputs. In this case, Our algorithm use small geometry to find the fully functional solutions, .the matrix has a size of 3×3. The resulting circuits as shown in Fig. 6. From the figures we know it is a gratifying result to obtain as it is clear that this design is an optimum solution.

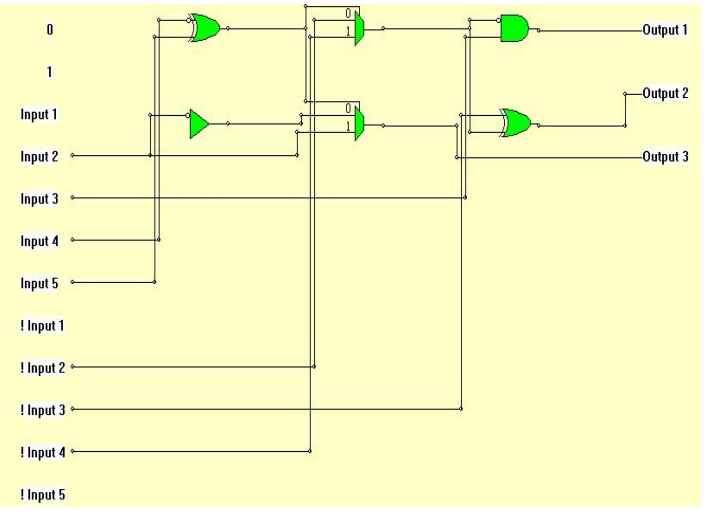


Fig. 6. The evolved optimum two-bit full adder circuit

4 Conclusion

Evolutionary Electronics applies the concepts of genetic algorithms to the evolution of electronic circuits. The main idea behind this research field is that each possible electronic circuit can be represented as an individual or a chromosome of an evolutionary process, which performs standard genetic operations over the circuits. In this paper we proposed a new means for designing electronic circuits given a set of logic gates. The final circuit is optimized in terms of complexity (with the minimum number of gates). For the case studies this means has proved to be efficient, experiments results show that we have attained the better results.

Acknowledgements

This paper is supported by Astronautics Research Foundation of China (NO. C5220060318).

References

1. Zebulum, R.S., Pacheco, M.A., Vellasco, M.M.: Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms. CRC Press, Boca Raton (2001)
2. Zebulum, R., Pacheco, M., Vellasco, M.: Variable Length Representation in Evolutionary Electronics. Evolutionary Computation 8(1), 93–120 (1997)
3. Thompson, A.: Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution. D. Phil. thesis, University of Sussex, Brighton, Sussex, England (1996)
4. Esbensen, H., Mazumder, P.: SAGA: A Unification of the genetic algorithm with simulated annealing and its application to macro-cell placement. In: Proceedings of the VLSI Design Conference, pp. 211–214. IEEE Computer Society Press, Los Alamitos (1994)
5. Esbensen, H.: A Macro-cell global router based on two genetic algorithms. In: Proceedings of the European Design Automation Conference, pp. 428–433. IEEE Computer Society Press, Los Alamitos (1994)
6. Kommu, V., Pomenraz, I.: GAFAP: Genetic Algorithm for FPGA technology mapping. In: European Design Automation Conference, pp. 300–305. IEEE Computer Society Press, Los Alamitos (1993)
7. Miller, J.F., Thomson, P.: Combinational and Sequential Logic Optimization Using Genetic Algorithms. In: Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 1995), IEE Conference Publications No. 414, London, England, pp. 34–38 (1995)
8. Miller, J.F., Thomson, P., Fogarty, T.: Designing Electronic Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study. In: Quagliarella, D., Periaux, J., Poloni, C., Winter, G. (eds.) Genetic Algorithms Recent Advancements and Industrial Applications, ch. 6. Wiley, New York (1997)
9. Higuchi, T., et al.: Evolvable Hardware with Genetic Learning. In: Proceedings of Simulated Adaptive Behavior. The MIT Press, Cambridge (1992)

10. Miller, J.F., Thomson, P.: Cartesian Genetic Programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 121–132. Springer, Heidelberg (2000)
11. Yan, X.S., Wei, W., et al.: Design Electronic Circuits by Means of Gene Expression Programming. In: Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems, pp. 194–199. IEEE Press, Los Alamitos (2006)

A Novel Particle Swarm Optimization for Constrained Engineering Optimization Problems

Minghai Jiao[1] and Jiafu Tang[2]

[1] The Computer Center, Northeastern University, Shenyang 110004, P.R. China
[2] Key Laboratory of Integrated Automation of Process Industry of Ministry of Education, Northeastern University, Shenyang 110004, P.R. China
mhjiao@cc.neu.edu.cn

Abstract. A novel particle swarm optimization algorithm CEOPSO (Constrained Engineering Optimization via Particle Swarm Optimization) is proposed for solving the constrained engineering design optimization problems in this paper. In CEOPSO, a new constraint handling mechanism with three rules of ε comparison technique is introduced to keep a certain proportion of infeasible individuals in the population during the iterations of PSO since some infeasible individuals that have better fitness values maybe promising. In addition, we present an improved doubly-link ring neighbourhood structure and a BLX-α combination method to prevent the search from being trapped into local minima. Computational results based on several well-known constrained engineering design problems show that the proposed CEOPSO algorithm performs better than the other recent approaches reported in the literature and demonstrate the robustness and competitiveness of CEOPSO.

Keywords: Swarm intelligence, Local PSO, Constraint handling.

1 Introduction

Constrained optimization problems, especially nonlinear optimization problems in which the objective function is minimized under some given constraints, are defined as the follows:

$$\text{Minimize } f(X),\ X = \{x_1, x_2, \ldots, x_n\} \in R^n \tag{1}$$

subject to:

$$g_i(X) \le 0, \qquad i = 1, 2, \ldots, p \tag{2}$$

$$h_j(X) = 0, \qquad j = 1, 2, \ldots, m \tag{3}$$

$$x_d^L \le x \le x_d^U, \qquad d = 1, 2, \ldots, n \tag{4}$$

where $X = \{x_1, x_2, \ldots, x_n\}$ is an n dimensional position vector, $f(X)$ is an objective function, $g_i(X) \le 0$ is p inequality constraints, $h_j(X) = 0$ is m equality constraints respectively. Values x_d^L and x_d^U are the lower and upper boundaries of x_i respectively.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 79–88, 2008.
© Springer-Verlag Berlin Heidelberg 2008

As equality constraints are difficult to be tackled in the optimization problems, a commonly used method is to transform the equality constraints into inequality ones based on the following inequality:

$$|h_j(x)| - \delta \leq 0, \qquad j = 1, 2, \ldots, m \tag{5}$$

where δ serves as a tolerance is a very small positive value.

Until now, the approaches for solving constrained optimization problems can be roughly classified into three categories in general. They are penalty, repair and tournament selection methods, respectively. The penalty method adds a penalty to the objective function to transform the constraint optimum problem to unconstraint problem [1]. Repair method applies operators to move an infeasible solution closer to the feasible space of solutions [2]. And the tournament selection method is based on the idea that any individual in a constrained search space must first comply with the constraints and then with the function value [3]. In recent years, many evolutionary algorithms have been proposed for solving constrained engineering optimization problems [4-8]. Especially, particle swarm optimization (PSO) algorithm as one of the recent meta-heuristics has gained increasing attention in tackling complex constrained optimization problems. PSO is a population-based stochastic optimization algorithm, firstly introduced by Kennedy and Eberhart in 1995 [9]. Nowadays many evolution strategies are incorporated into PSO algorithm for tackling constrain engineering optimization problems, such as struggle selection method [10], addition of ranking method [11], co-evolutionary PSO approach [12] and so on [13-15].

In this paper, we present a novel approach based on PSO algorithm to solve the constrained engineering design optimization problems. Some strategies including an improved doubly-link ring neighbourhood structure and a BLX-α combination method are used to prevent PSO from being premature convergence. Moreover, we also propose a new constraint handling mechanism to explore some promising search space on the boundary between the feasible and the infeasible regions.

This paper is organized as follows: details of global and local PSO are described in Section 2. In Section 3, we discuss three key components of CEOPSO. Simulation results and comparisons with previously reported results are presented in Section 4. Finally, this paper ends with some conclusions and future works in Section 5.

2 Global and Local PSO

In global PSO, all the particles in the swarm search the real valued n-dimensional search space for finding the optimum position. The searching procedures are iterative, and the position and velocity are used to explore and exploit the optimum solution in the search space. At any time t, each particle knows its current position x_i^t and remembers its personal best position x_{iPBest}^t until now. Moreover, every member in the flock knows the global best position x_{GBest}^t which represents the best particle in flock. The velocity v_i^{t+1} of particle i can be calculated according to the following equation:

$$v_i^{t+1} = w v_i^t + c_1 r_1 (x_{iPBest}^t - x_i^t) + c_2 r_2 (x_{GBest}^t - x_i^t) \tag{6}$$

where c_1 and c_2 are constants called *acceleration coefficients*, the parameters r_1 and r_2 are two independent random numbers in the interval [0, 1]. The parameter w is called *inertia weight*. Then, the position x_i^{t+1} of particle i is updated in each generation as shown bellow:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{7}$$

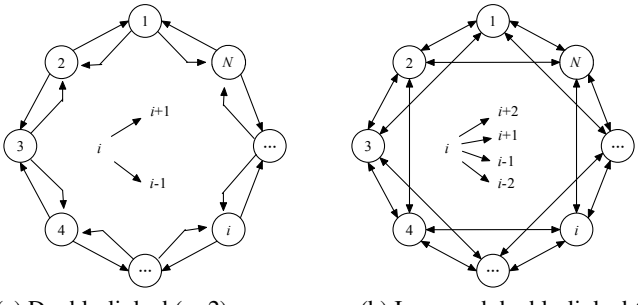
In local PSO, the leader of local PSO which is denoted as x_{iLBest}^t is the best member of a flock neighbourhood. So, instead of equation (6), the velocity of particle i in local PSO can be computed as follows:

$$v_i^{t+1} = wv_i^t + c_1 r_1 (x_{iPBest}^t - x_i^t) + c_2 r_2 (x_{iLBest}^t - x_i^t) \tag{8}$$

3 Particle Swarm Optimization for CEOPSO

3.1 Improved Doubly-Linked Ring Neighbourhood Structure

The ring doubly-linked neighbourhood structure is composed of N bidirectional search particles where each particle is assigned to a permanent label used to construct neighbourhoods. The ring neighbourhood structure must be linked end to end, such as particle 1 and particle N, which are immediate neighbours. The doubly-linked list is generally composed by n neighbours and size n is 2 as shown in Fig. 1-a. For example, particle i has 2 neighbours: particle i+1 and particle i-1, and the best particle among the three particles i-1, i, i+1 is the local best *LBest* in neighbourhood of particle i.



(a) Doubly-linked (n=2) (b) Improved doubly-linked (n=4)

Fig. 1. Ring doubly-linked neighborhood structure

We propose an improved doubly-linked list of ring neighbourhood structure in order to promote communication among neighbourhoods and obtain more precise local information than those of ring doubly-linked neighbourhood structure (n=2). The neighbourhood size n is set to 4 in the new neighbourhood structure which is shown in Fig. 1-b. Here, particle i has 4 neighbors: particle i+1, particle i+2, particle i-1 and particle i-2. The details of this new neighbourhood structure can be explained as follows. Suppose that particle 1 is the best of particle 2's neighbourhood since particle 2

has particle 1, particle *N*, particle 3 and particle 4 as neighbors. Then, particle 2 is attracted by particle 1. Simultaneously, particle 2 is the leader in the neighbourhood of particle 3 whose neighbourhood includes particles 1-5. So, particle 3 is directly pulled by particle 2 and is influenced indirectly by particle 1. Similarly, the other particles in the swarm evolve in the same way. Finally, such a neighbourhood structure makes the entire search procedure explore and exploit towards the best position in the search space.

3.2 Constraint Handling Mechanism

CEOPSO converts the constrained optimization problem into a bi-objective optimization problem including the minimal fitness function *Fitness*(x) and constraint violation function $\Phi(x)$. The constraint violation function $\Phi(x)$ is defined as the accumulative sum of the constraint violation, which is composed of the violation of both inequality and equality constraint conditions, shown as equation (9). Clearly, moving the particles towards the feasible region is equivalent to optimize the bi-objective problem of free constraint.

$$\Phi(x) = \sum_{i=1}^{p} \max\{0, g_i(x)\} + \sum_{j=1}^{m} \max\{0, |h_j(x)| - \delta\} \tag{9}$$

The popular tournament selection rules for local best *LBest* or personal best individual *PBest* in search space are as follows: (1) Given two feasible particles, pick the one with better fitness function value;(2) If both particles are infeasible, pick the particles with the lowest constraint violation value of $\Phi(x)$; (3) From a pair of feasible and infeasible particles, pick the feasible one.

For general constrained optimization problem, some infeasible individuals maybe exist near the global optimum and hold good fitness values. Although they are infeasible in current iteration, further operations maybe make them to create new feasible offspring with better fitness value. To some extent, it is helpful for the optimization to keep one small part of infeasible but good individuals. Thus, we propose a constant ε as the constraint violation tolerance near boundary margin of feasible region and infeasible region in the propose PSO. For a predefined constants ε (ε>0), the tournament-selection comparison rules for a pair of feasible or infeasible particles are improved as the follows:

a) If both constraint violation sums of particles are smaller than or equal to ε ($\Phi(x) \leq \varepsilon$), pick the one with better fitness function value;
b) If both constraint violation sums of particles are larger than ε ($\Phi(x) > \varepsilon$), pick the one with smaller $\Phi(x)$ value;
c) Given that one particle satisfies the constraint violation ε ($\Phi(x) \leq \varepsilon$) and the other particle does not ($\Phi(x) > \varepsilon$), the former is picked.

3.3 BLX-α Combination Method

The BLX-α combination method can be described as follows. Assume that $X=(x_1,\ldots,x_i,\ldots,x_n)$ and $Y=(y_1,\ldots,y_i,\ldots,y_n)$ are two parents. Then, an offspring is generated as $Z=(z_1,\ldots,z_i,\ldots,z_n)$, where z_i is a random number in the interval $[c_{\min}\text{-}I\cdot\alpha,$

$c_{max}+I\cdot\alpha$] and $c_{max}=\max(x_i, y_i)$, $c_{min}=\min(x_i, y_i)$, $I = c_{max} - c_{min}$. Constant α is a combination factor chosen from interval [0, 1] and is often set to 0.5 as recommended by [16].

CEOPSO uses BLX-α combination method as a diversity strategy to add diversity to the swarm. Considering that the global best particle *GBest* and the best infeasible particle *BInfeas* are the two most promising particles in the swarm, we merger two kinds of BLX-α combination in CEOPSO. One is a combination between *GBest* and the personal best particle *PBest*. And the other is a combination between *BInfeas* found so far and *PBest*. A probability p serves as a threshold value of the two kinds of BLX-α combination where p is defined as the ratio of the number of the infeasible particles to the number of the entire population. At each generation, a number r is generated randomly from [0, 1]. If r is less than p, then the combination between *BInfeas* and *PBest* occurs for each particle in the swarm. Otherwise, *GBest* and *PBest* are combined to generate an offspring for every particle.

3.4 The Framework of Proposed CEOPSO

At each generation, the framework of proposed CEOPSO algorithm mainly consists of two stages. In the first stage, the local PSO with an improved ring doubly-linked neighbourhood structure is implemented. Then, in the second stage, the BLX-α combination method is applied to generate offspring particles variation. The particle tournament selection is based on the constraint handling mechanism with ε as the constraint violation tolerance. Let U be the upper boundary of particle position and L be the lower boundary of particle position. The pseudo-code of CEOPSO algorithm is in the following:

```
  For i=1 To N
     X_i^0 = Rand(L, U)
     V_i^0 = Rand(-(U-L)/2,(U-L)/2)
     Fit_i^0 = Fitness(X^0)
     Vio_i^0 = ( X_i^0 )
     X_iPBest^0 = X_i^0
     FitBest_i^0 = Fit_i^0
     VioBest_i^0 = Vio_i^0
  End for
  For t=1 To Maxgenerations
stage1
    For i=1 To N
       X_iLBest^t-1 =LocalBest( FitBest_i^t-1 , VioBest_i^t-1 ,NeighborSize,ε)
       V_i^t = Velocity( V_i^t-1 , X_i^t-1 , X_iPBest^t-1 , X_iLBest^t-1 )
       X_i^t = X_i^t-1 + V_i^t
       Fit_i^t = Fitness( X_i^t )
       Vio_i^t = Φ( X_i^t )
```

```
      Update( X^t_iPBest , FitBest^t_i , VioBest^t_i , X^t_GBest )
    End for
stage2
    r = Rand(0, 1)
    IF(r < p)
      For i=1 To N
        BLX-α( X^t_iPBest , X^t_BInfeas )
      End For
    End IF
    If(r ≥ p)
      For i=1 To N
        BLX-α( X^t_iPBest , X^t_GBest )
      End For
    End If
  End For
```

4 Experiments and Discussions

4.1 Experimental Setup

A set of four well-known constrained engineering design optimum problems are carried out to evaluate the performance of CEOPSO:

(1) Problem E01: Design of a Welded Beam
(2) Problem E02: Design of a Pressure Vessel
(3) Problem E03: Minimization of the Weight of a Tension/Compression Spring
(4) Problem E04: Himmelblau's Nonlinear Optimization Problem.

The parameter settings of CEOPSO are described as follows: N=30, *NeighborSize*=4, c_1=c_2=1.49445, α=0.5, *Maxgenerations*=1000, w is linearly decreased from 0.9 to 0.4. The maximum positions $X_{\max}$ and minimum positions $X_{\min}$ of particles depend on the variable region of problems. The maximum and minimum velocities for particles are set as $V_{\max}$=($X_{\max}$-$X_{\min}$)/2 and $V_{\min}$= $-V_{\max}$, and the initial velocity of particle i is set as V_i^0 =*rand*×$V_{\max}$ -0.5×$V_{\max}$, where *rand* is a random number from interval [0, 1]. CEOPSO is run 30 times independently.

The parameter sensitive analysis on ε tolerance value is shown in Fig. 2. The fitness values during searching process are pictured when the parameter ε is set to 0.01, 1, 100 respectively. The results show that CEOPSO performs well when the value of parameter ε is equal to 1 and 0.01. Thus, we set ε=1 in all the experiments for CEOPSO.

4.2 Stimulation Results and Discussion

We compare CEOPSO with standard PSO algorithms and some other evolutionary algorithms with different constraint handling mechanisms [4, 6, 7, 8, 10, 12, 18, 19, 20, 21]. Tables 1-4 report the computational results on E01- E04 in detail.

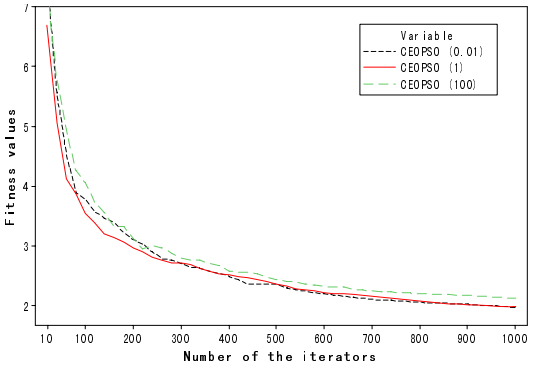


Fig. 2. Comparison of different values on parameter ε

Take welded beam design problem (E01) as an example, the fitness values found by CEOPSO and standard PSO including both local PSO and global PSO during the iterations are shown as Fig. 3. From the best fitness trend-lines in Fig. 3, we can find that CEOPSO is more efficient than standard PSO algorithms.

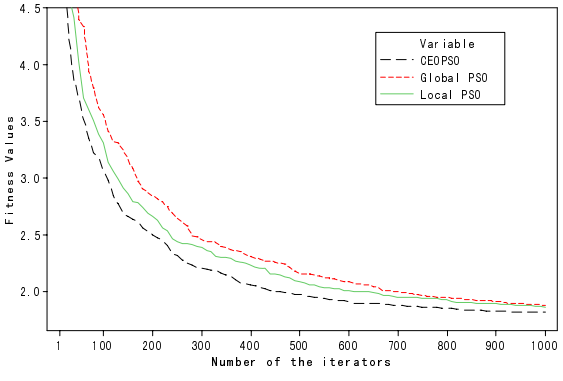


Fig. 3. Best fitness trend-lines of CEOPSO and standard PSO on E01

From Table 1 we compare the results of CEOPSO with Deb's GA algorithm, Co-evolution GA algorithm, CPSO algorithm, Ragsdell's algorithm, and Dominance GA algorithm for the problem E01. It can be seen that the best feasible solution found by CEOPSO is better than the best solutions found by other algorithms. In addition, the standard deviation of CEOPSO in 30 independent runs is very small.

From Table 2, the problem E02 is used to compare the results of CEOPSO with GeneAS algorithm, Co-evolution GA algorithm, CPSO algorithm, Sandgren's algorithm, Kannan and Kramer's algorithm, and Dominance GA algorithm. It can be seen that the average searching quality of CEOPSO is also better than those of other algorithms, and even the worst solution found by CEOPSO is better than the best solutions found by the other compared algorithms.

Table 1. Comparison with the different algorithms for problem E01

Algorithm	Best	Mean	Worst	Std Dev
CEOPSO	1.724850	1.724956	1.725910	0.000318
Deb's GA[4]	2.433116	N/A	N/A	N/A
Co-evolution GA[7]	1.748309	1.771973	1.785835	0.011220
CPSO[12]	1.728024	1.748831	1.782143	0.012926
Ragsdell and Phillips[17]	2.385937	N/A	N/A	N/A
Dominance GA[21]	1.728226	1.792654	1.993408	0.074713

N/A—data not available.

From Table 3, the problem E03 is used to compare the results of CEOPSO with Belegundu's algorithm, Arora's algorithm, Co-evolution GA algorithm, Dominance GA algorithm, and CPSO algorithm. Also, CEOPSO outperforms the other algorithms in terms of solution quality.

Table 2. Comparison with the different algorithms for problem E02

Algorithm	Best	Mean	Worst	Std Dev
CEOPSO	5885.3300	5908.6700	6008.8200	39.2047
GeneAS[6]	6410.3811	N/A	N/A	N/A
Co-evolution GA[7]	6288.7445	6293.8432	6308.1497	7.4133
CPSO[12]	6061.0777	6147.1332	6363.8041	86.4545
Sandgren[18]	8129.1036	N/A	N/A	N/A
Kannan and Kramer[20]	7198.0428	N/A	N/A	N/A
Dominance GA[21]	6059.9463	6177.2533	6469.3220	130.9297

Table 3. Comparison with the different algorithms for problem E03

Algorithm	Best	Mean	Worst	Std Dev
CEOPSO	0.0126652	0.012705	0.0127191	1.659712e-005
Belegundu[23]	0.0128334	N/A	N/A	N/A
Arora[19]	0.0127303	N/A	N/A	N/A
Co-evolution GA[7]	0.0127048	0.012769	0.012822	3.939000e-005
Dominance GA[21]	0.0126810	0.0127420	0.012973	5.900000e-005
CPSO[12]	0.0126747	0.012730	0.012924	5.198500e-005

Table 4 reports the experimental results of problem E04. We compare CEOPSO with the other algorithms with respect to the best feasible solution and find that the result of CEOPSO is almost the same as that of COPSO and Hu's algorithm.

Table 4. Comparison with the different algorithms for problem E04

Algorithm	Best	Mean	Worst	Std Dev
CEOPSO	-31025.600000	N/A	N/A	N/A
Co-evolution GA[7]	-31020.859000	N/A	N/A	N/A
COPSO[8]	-31025.560242	N/A	N/A	N/A
Hu and Eberhart[15]	-31025.561420	N/A	N/A	N/A
Homaifar[24]	-30665.609000	N/A	N/A	N/A

5 Conclusions

This paper proposes a novel CEOPSO algorithm which has shown high performance in constraint engineering optimization problem of nonlinear nature. This is the first report to incorporate an improved doubly-linked list of ring neighbourhood structure into PSO to solve constrained optimization problem. Two important contributions of CEOPSO are also worth to mention: the ε comparison rules for tournament selection between feasible and infeasible particles, and BLX-α combination method as an effective diversity strategy that prevents the swarm from being trapping into local optima to some extent. Simulation results based on some well-known constrained engineering design problems provide evidence that CEOPSO is highly competitive and robust. The future work will improve the ε comparison rules to incorporate a dynamic ε tolerance into CEOPSO in order to further enhance exploration and exploitation abilities. In addition, the doubly-linked ring neighbourhood structure of local PSO will be further improved for shorting the running time of algorithms.

Acknowledgments. The paper is financially supported by the Natural Science Foundation of China (Grant No. 70625001, 70721001, 70471028), the program of New Century Excellent Talents in University (NCET-04-280) of MOE of China, and 111 project of Ministry of Education (MOE) in China with number B08015.

References

1. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Reading (1989)
2. Davis, L.: Genetic Algorithms and Simulated Annealing. Pitman, London (1987)
3. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186, 311–338 (2000)
4. Deb, K.: Optimal Design of a Welded Beam via Genetic Algorithms. AIAA Journal 29, 2013–2015 (1991)
5. Michalewicz, Z.: A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. In: McDonnell, J.R., et al. (eds.) Proceedings of the fourth Annual Conference on Evolutionary Programming, pp. 135–155. MIT Press, Cambridge (1995)
6. Deb, K.: GeneAS: a Robust Optimal Design Technique for Mechanical Component design. In: Dasgupta, D., Michalewicz, Z. (eds.) Evolutionary Algorithms in Engineering Applications, pp. 497–514. Springer, Berlin (1997)
7. Coello, C.A.C.: Use of a Self-adaptive Penalty Approach for Engineering Optimization Problems. Computers in Industry 41, 113–127 (2000)
8. Aguirre, A.H., Zavala, A.E.M., Diharce, E.V., Rionda, S.B.: COPSO: Constrained Optimization via PSO Algorithm. Technical Report. I-07-04, Center of Research in Mathematics (CIMAT), Guanajuato, Mexico (2007)
9. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of the 1995, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
10. Coello, C.A.C.: Theoretical and Numerical Constraint Handling Techniques Used with Evolutionary Algorithms: A Survey of the State of the Art. Computer Methods in Applied Mechanics and Engineering 191, 1245–1287 (2002)

11. Ho, P.Y., Shimizu, K.: Evolutionary Constrained Optimization Using an Addition of Ranking Method and a Percentage-based Tolerance Value Adjustment Scheme. Information Sciences 177, 2985–3004 (2007)
12. He, Q., Wang, L.: An Effective Co-evolutionary Particle Swarm Optimization for Constrained Engineering Design Problems. Engineering Applications of Artificial Intelligence 20, 89–99 (2007)
13. Runarsson, T.P., Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation 4, 284–294 (2000)
14. Eberhart, R.C., Shi, Y.: Tracking and Optimizing Dynamic Systems with Particle Swarms. In: Proceedings of the IEEE International Congress on Evolutionary Computation, Seoul, Korea, pp. 94–97 (2001)
15. Hu, X.H., Eberhart, R.C., Shi, Y.H.: Engineering Optimization with Particle Swarm. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS 2003), pp. 53–57. IEEE Neural Networks Society, Indianapolis (2003)
16. Deb, K.: A Population-based Algorithm-generator for Real-parameter Optimization. KanGAL Report. No. 2003003, Kanpur Genetic Algorithms Laboratory, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur (2003)
17. Ragsdell, K.M., Phillips, D.T.: Optimal Design of a Class of Welded Structures Using Geometric Programming. ASME Journal of Engineering for Industries 98, 1021–1025 (1976)
18. Sandgren, E.: Nonlinear Integer and Discrete Programming in Mechanical Design. In: Proceedings of the ASME Design Technology Conference, Kissimine, FL, pp. 95–105 (1988)
19. Arora, J.S.: Introduction to Optimum Design. McGraw-Hill, New York (1989)
20. Kannan, B.K., Kramer, S.N.: An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and its Applications to Mechanical Design. Transactions of the ASME. Journal of Mechanical Design 116, 318–320 (1994)
21. Coello, C.A.C., Montes, E.M.: Constraint-handling in Genetic Algorithms through the Use of Dominance-based Tournament Selection. Advanced Engineering Informatics 16, 193–203 (2002)
22. Hu, X.H., Eberhart, R.C., Shi, Y.H.: Engineering Optimization with Particle Swarm. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS 2003), pp. 53–57. IEEE Neural Networks Society, Indianapolis (2003)
23. Belegundu, A.D.: A Study of Mathematical Programming Methods for Structural Optimization. Department of Civil and Environmental Engineering, University of Iowa, Iowa City, Iowa (1982)
24. Homaifar, A., Lai, S.H.Y., Qi, X.: Constrained Optimization via Genetic Algorithms. Simulation 62, 242–254 (1994)

An Improved Differential Evolution Based on Triple Evolutionary Strategy*

Yichao He[1], Yingzhan Kou[2], and Chunpu Shen[3]

[1] School of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang, 050031, China
heyichao@sjzue.edu.cn
[2] Computer Engineering Department, Ordnance Engineering College, Shijiazhuang, 050003, China
kouyingzhan@yahoo.com.cn
[3] College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang, 050016, China

Abstract. Differential Evolution (DE) is an evolution algorithm that was proposed by Storn and Price in 1997, which has already succeeded in applying to solve optimization questions in a lot of fields. This paper discusses various kinds of characteristics that DE demonstrates at first, then propose an improved differential evolution algorithm (TSDE) which has three kinds of efficiently evolutionary strategies, and proves its global convergence property use the finite Markov chain theory. Through the compare result of calculation of 5 classics testing functions, it shows that TSDE has the obvious advantage in the quality of solution, adaptability, robustness etc. than original DE and DEfirDE.

1 Introduction

Differential Evolution(DE)[1,2] is a novelty evolutionary algorithm which was proposed by Rainer Storn and Kenneth Price, it has gain much progress about DE. For example, literature[3] described a pareto DE algorithm which is adaptive automatic, literature[4] use DE to solve multi-objective optimization problem, literature[5] proposed a binary DE algorithm to solve combinatorial optimization problem, literature[6] combine local search and one-point crossover and proposed an improved DE algorithm to solve high dimensional function optimization, literature[7] proposed an improved DE algorithm based on mixed optimization strategy. It has successfully used in optimization function, training nerve network, electromagnetism, system control and other fields[8-10].

This paper introduces the principle and analyze characteristics of DE in section 2. In section 3 and 4,We propose an improved differential evolution (TSDE), and prove its convergence use random Markov chain theory. Afterwards, through the compare result of 5 classics testing functions, it shows that TSDE has the obvious advantage in the quality of solution, adaptability, robustness and etc than original DE. Finally, summarize and propose the direction of research in the future.

* This paper is supported by the National Natural Science Foundation of China under Grant (NO. 60471022) and Key Scientific Research project of Hebei Province (NO.07216926).

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 89–97, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Differential Evolution and Its Characteristics

DE is a evolution algorithm based on coding real number. In every generation of evolution, first use the Differential Operator(DO) that defined below to get a middle population make up of mixed individual of father and son, and then use Select Operator(SO) to recombine a new population use the strategy of winner stay but losser leave. Now use DE/*r*/1/*bin* evolutionary strategy of DE to introduce the general principle of this algorithm.

We suppose the problem to solve is the minimum optimization. the target function $f(x)$ has d variable, search space $S=\{X|X=(x_1,x_2,\ldots,x_d)\wedge L_i\le x_i\le U_i,\ i=1,2,\ldots,d\}$, L_i, $U_i\in R$. Suppose t is the current iterativeness of algorithm, s is magnitude of the population, vector $X_i=(x_{i1},x_{i2},\ldots,x_{id})$ is the individual of number i in the population, vector $V_i=(v_{i1},v_{i2},\ldots,v_{id})$ is the individual of number i in the middle population, $i=1,2,\ldots,s$, and then the DO of differential evolution is

$$v_{ij}(t+1)=\begin{cases}x_{p_1j}(t)+\alpha(x_{p_2j}(t)-x_{p_3j}(t)), & if\ r\le CR\quad or\quad j=R(i)\\ x_{ij}(t), & others.\end{cases}\tag{1}$$

In them, $v_{ij}\in V_i$, x_{ij}, $x_{p1,j}$, $x_{p2,j}$, $x_{p3,j}\in X_i$, and $p_1\neq p_2\neq p_3\neq i$, $1\le i\le s$, $1\le j\le d$. $\alpha<1$ is a real and constant factor, r is a random number and $r\sim(0,1)$; CR is called cross probability, and $CR\in[0,1]$, $R(i)$ is a random positive integer in $[1, n]$, the SO of differential evolution is

$$X_i(t+1)=\begin{cases}V_i(t+1), & if\quad f(V_i(t+1))<f(X_i(t))\\ X_i(t), & others.\end{cases}\tag{2}$$

In every iterativeness of DE evolution, we suppose X_{best} is the best individual in current population, MAXT is the maximum iteration, and then DE algorithm[2,5] can be described as follows:

DE Algorithm

1. Randomly generate original population $P(0)=\{X_i|1\le i\le s\}$,and set $T=1$;
2. While ($T\le$ MAXT) Do
3. For $i=1$ to s Do
4. For $j=1$ to d Do
5. Randomly generate $r\in(0,1)$ and $R(i)\in[1, d]$;
6. if ($r\le CR\vee j=R(i)$) then $v_{ij}=x_{p1,j}+a(x_{p2,j}-x_{p3,j})$ else $v_{ij}=x_{ij}$;
7. if $f(V_i)<f(X_i)$ then $X_i=V_i$;
8 Caculate the most optimization in $P(T)$ as $X_{best}(T)$, set $T=T+1$;
9. EndWhile
10. Output $X_{best}(T)$ and $f(X_{best}(T))$, end the algorithm.

Through the description of the algorithm, we know the time complication of DE iterate one time is $O(N^2)$, in them $N=\max\{s, d\}$.

DE has two basic characteristics of general algorithm: implicit parallelism and use effectively the global information[11]. Besides this, it also has the following characteristics: (1) DE has the characteristic of mutation. Base on the definition of DO operator, $X_i(t)$ and $X_{p1}(t)+\alpha(X_{p2}(t)-X_{p3}(t))$ contribute corresponding son variable and

generate middle individual $V_i(t+1)$, from the point of view of evolution course, its essence is to mutate in the probability of $CR+1/d$ in every dimensional of basic individual $X_i(t)$ and generate new individual $V_i(t+1)$, the middle individual of middle population in the $t+1$ generation is mutated in the probability of $1-(1-CR-1/d)^2$ in the t generation. (2) DE has characteristic of learning. On average of $CR \times d+1$ son variables in the middle individual of $V_i(t+1)$ come from $X_{p1}(t)+\alpha(X_{p2}(t)-X_{p3}(t))$, it shows that $V_i(t+1)$ come from the learning of $X_{p1}(t)$, $X_{p2}(t)$ and $X_{p3}(t)$, it indicates the characteristic of learning. (3) DE has the characteristic of freedom exploration. In DO operator, individual $X_{p1}(t)$, $X_{p2}(t)$, $X_{p3}(t)$ were chosen randomly and unconstrainedly, so $V_i(t+1)$ which is formed by them is unconstrained and random. This indicates that DO operator produces the new individual by way of unconstrained freedom, it has reflected the free exploration of DE.

3 Improved DE with Triple Evolutionary Strategy

The evolution strategy of DE expressed as DE/*x*/*y*/*z*[6] generally. literature[6,7] first notice performance difference in various kinds of evolution strategy of DE, but this paper first analysis the performance difference of the evolution strategy of DE, and then point out the generality characteristic of all strategy, and then propose an improved, concise and high-efficient algorithm based on the general characteristic of three strategies and performance difference.

To depict conveniently, we only discuss the evolution strategy based on the way of binomial recombination in the following. All of the conclusions for the exponential recombination way are still tenable. In all kinds of evolution strategies of DE, we notice that its basic structure is: the new individual V_i is produced by benchmark individuals(that is X_i, X_{p1} or X_{best})and differential vector recombination. As thus, because the different choosing between the benchmark individuals and differential vectors, the performances embodied by each evolution strategy are also different. Among them, the DE/*rand*/1/*bin* strategy chooses the benchmark individuals and differential vectors at random, and the freedom exploration of its DO is more outstanding, and its global convergence is relatively better. DE/*best*/1/*bin* strategy use X_{best} of the present population as the benchmark individual, producing the middle individual by X_{best} and differential vector choused at random, and local convergence of its DO is relatively good, and speed of convergence is rather fast. DE/*rand*-to-*best*/1/*bin* strategy produce new individual by X_i, (X_{best} - X_i) and (X_{p1}- X_{p2}). Its the freedom exploration and local convergence is relatively equilibrium, and have better adaptability.

From the basic structure of each evolution strategy, it can be find out that they have two generality characteristics: (1) the combination ways of producing the new individual of each strategy are the same, and all of they are linear mode of combinations by the benchmark individuals and differential vectors. (2) The DO's formula of each strategy at least has a differential vectors chosen at random. When many evolution strategies are used at the same time, it can be naturally realized the information transformation and sharing among the strategies through choosing differential vectors at random.

Based on the analysis above, we proposed Differential Evolution based on Triple Evolutionary Strategy (for short TSDE). In order to stress the freedom exploration as

well as keep relatively equilibrium of global and local convergence of the algorithm, the three especial strategies above are used in TSDE. On one hand, DE/*rand*/1/*bin* strategy with stronger freedom exploration can not merely make the algorithm have better global convergence, and also can improve the variety of the population. On the other hand, as DE/*rand*/1/*bin* strategy with stronger local convergence and DE/*rand*-to-*best*/1/*bin* with relatively equilibrium of freedom exploration and local convergence, they can accelerate the speed of convergence of the algorithm on the basis of keeping the population diversified. In this way, relying mainly on DE/*rand*/1/*bin* strategy, and making DE/*best*/1/*bin* and DE/*rand*-to-*best*/1/*bin* strategy subsidiary, three kinds of strategies cooperate and evolve together each other. Keeping equilibrium of global and local convergence as well as paying attention to freedom exploration and trend towards the excellent effective combination, it can make TSDE algorithm have higher convergence property.

In order to describe TSDE algorithm, TSDO is redefined on the basis of Do and triple evolutionary strategy as follows.

Define 1. Suppose that vector $X_i=(x_{i1},x_{i2},\ldots,x_{id})$ is the ith individual of the tth generation of middle population, and vector $V_i=(v_{i1},v_{i2},\ldots,v_{id})$ is the ith individual of the (t+1)th generation of population (i=1,2,…,s, $Rand\in(0,1)$ is random variable). Then, Triple Evolutionary Strategy Differential Operator (TSDO) relying mainly on DE/*rand*/1/*bin* strategy is that

$$v_{ij}(t+1)=\begin{cases} x_{p_1,j}(t)+\alpha(x_{p_2,j}(t)-x_{p_3,j}(t)), & if\ (r\le CR\vee j=R(i))\wedge(Rand\le 0.5)\\ x_i(t)+\alpha(x_{best,j}-x_{ij})+\alpha(x_{p1,j}-x_{p2,j}), & if\quad (r\le CR\vee j=R(i))\wedge(0.5<Rand\le 0.8)\\ x_{best,j}(t)+\alpha(x_{p1,j}(t)-x_{p2,j}(t)), & if\ (r\le CR\vee j=R(i))\wedge(rand>0.8)\\ x_{ij}(t), & others. \end{cases} \tag{3}$$

other parameters in the formula (3) are the same as DO's.

On the basis of Define 1, pseudo C-code of TSDE algorithm combined SO is described as follows:

TSDE Algorithm

1. Produce the initial population at random $P(0)=\{X_i\,|1\le i\le s\}$, and set T=1;
2. While ($T\le$MAXT) Do
3. For i=1 to s Do
4. For j=1 to d Do
5. Produce at random $r\in(0,1)$ and $R(i)\in[1,d]$;
6. if ($r>CR\wedge j\neq R(i)$) then $v_{ij}=x_{ij}$ and Continue iteration;
7. Produce at random $Rand\in(0,1)$;
8. if $Rand\le 0.5$ then $v_{ij}=x_{p1,j}+\alpha(x_{p2,j}-x_{p3,j})$;
9. if $0.5<Rand\le 0.8$ then $v_{ij}=x_{ij}+\alpha(x_{best,j}-x_{ij})+\alpha(x_{p1,j}-x_{p2,j})$;
10. if $Rand>0.8$ then $v_{ij}=x_{best,j}+\alpha(x_{p1,j}-x_{p2,j})$;
11. if $f(V_i)<f(X_i)$ then $X_i=V_i$;
12. Calculate $X_{best}(T)$ of $P(T)$, and set $T=T+1$;
13. EndWhile
14. Output $X_{best}(T)$and $f(X_{best}(T))$, end the algorithm.

In TSDE, as only one step of the 6th and 8th to 10th is executed each iterative, TSDE as to DE does not increase extra calculation, and the time complexity of each iterative

is still $O(N^2)$ and N=max{s, d}. To facilitate comparing, DE which based on DE/*rand*/1/*bin*, DE/*best*/1/*bin* and DE/*rand*-to-*best*/1/*bin* strategy is calld as DE1, DE3 and DE5 in the following respectively.

4 Prove the Global Convergence of TSDE

Actually, suppose that the precision that need to calculate is $\varepsilon=10^{-k}$, then the individual's every element in DE is a real number with k valid figures after the decimal point; so that the search space $S=\prod_{i=1}^{d}[L_i,U_i]$ is the discrete space formed with the whole d-dimensional real vectors with k valid figures after the decimal point (Here L_i<U_i, L_i and U_i are the real number). The size of search space is $|S|=\prod_{i=1}^{k}[(U_i-L_i)10^k]\le[10^k(U-L)]^d$, and U=max{ $\overline{U}_i \mid \overline{U}_i=\lceil U_i\rceil \wedge 1\le i\le d\}$, L=min{ $\overline{L}_i \mid \overline{L}_i=\lfloor L_i\rfloor \wedge 1\le i\le d\}$. The fitness value of each vector $X\in S$ is written for $fit(X)$. Set F={ $fit(X)\mid X\in S$}, obviously $|F|\le|S|$, then F can be expressed as F={F_1, F_2 ,…, $F_{|F|}$}, and F_1<F_2 <…<$F_{|F|}$. According to the difference of fitness value, divide S into $S=\bigcup\{S_i\}$, and S_i={$X\mid X\in S\wedge fit(X)=F_i\}$, i=1,2,…, $|F|$, then

$$\sum_{i=1}^{|F|}|S_i|=|S|,\quad S_i\neq\Phi,\ \forall i\in\{1,2,...,|F|\},\ S_i\cap S_j=\Phi,\ \forall i\neq j,\ \bigcap_{i=1}^{|F|}S_i=S.$$

Obviously, F_1 is the best global solution F*, and S_1 includes all individuals whose fitness value is F*.

In TSDE, the population writes for P={X_1,X_2,…,X_s}. Make **P** gather for all the populations. Because the individual allows the same in the population, the possible number of the populations is $|\mathrm{P}|=\binom{|S|+s-1}{s}$. In order to weigh the population, define the fitness value of population P as $fit(P)$=max{ $fit(X_i)\mid X_i\in P\wedge i$=1, 2,…,$s$}, then $F_1\le fit(P)\le F_{|F|}$, $\forall P\in\mathbf{P}$. Therefore, Set **P** can be divided into several non-null subset {$\boldsymbol{P}_i$}, $\boldsymbol{P}_i$= {$P\mid P\in\mathbf{P}\wedge fit(P)=F_i$}, i=1,2,…, $|F|$. And

$$\sum_{i=1}^{|F|}|\mathrm{P}_i|=|\mathrm{P}|,\ \mathrm{P}_i\neq\Phi,\ \forall i\in\{1,2,...,|F|\},\ \mathrm{P}_i\cap\mathrm{P}_j=\Phi,\ \forall i\neq j,\ \bigcap_{i=1}^{|F|}\mathrm{P}_i=\mathrm{P}.$$

Set $\boldsymbol{P}_1$ includes populations with the fitness value F*.

Make P_{ij} express the jth population of $\boldsymbol{P}_i$, i=1,2,…, $|F|$, j=1,2,…, $|\boldsymbol{P}_i|$. Under the function of TSDO, population's state transformed from P_{ij} to P_{kl} can be expressed for $P_{ij}\rightarrow P_{kl}$. Order $p_{ij,\,kl}$ shows the transformation probability from P_{ij} to P_{kl}, $p_{ij,\,k}$ expresses the transformation probability from P_{ij} to any population in $\boldsymbol{P}_k$, $p_{i,k}$ expresses the transformation probability from any population in $\boldsymbol{P}_i$ to any one in $\boldsymbol{P}_k$. Then, $p_{ij,k}=\sum_{l=1}^{|\mathrm{P}_k|}p_{ij,kl}$, $\sum_{k=1}^{|F|}p_{ij,k}=1$, $p_{i,k}\ge p_{ij,k}$.

After defining above-mentioned concepts, on the basis of [12], [13] and [14], TSDE's global convergence based on limited Markov chain theory will be proved in the following.

Define 2. Suppose that $P(t)$ is the tth evolution population of TSDE, and F^* is the optimum global fitness value, if

$$\lim_{t\to\infty} prob\{fit(P(t)) = F^*\} = 1 \tag{4}$$

is certainly, then call TSDE global convergence. Due to $F=F_1$, formula (4) is equivalent to $\lim_{t\to\infty} prob\{P(t)\in \mathrm{P}_i\} = 1$.

Lemma 1[13]**.** Suppose that P' is a reduced random matrix of an n rank, which can get $\mathrm{P}' = \begin{pmatrix} \mathrm{C} & 0 \\ \mathrm{R} & \mathrm{T} \end{pmatrix}$ after the same row and line transforms. Among them C is a primal random matrix of an m rank, and R, $T\neq 0$. Then

$$\mathrm{P}'^{\infty} = \lim_{n\to\infty} \mathrm{P}'^{k} = \lim_{n\to\infty} \begin{bmatrix} C^k & 0 \\ \sum_{i=0}^{k-1} T^i R C^{k-i} & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & 0 \\ R^{\infty} & 0 \end{bmatrix}$$

is a steady random matrix, and $\mathrm{P}'^{\infty} = e' \times p'^{\infty}$. Here $e=(1,1,\ldots,1)$, $p'^{\infty} = p'^{0} \times \mathrm{P}'^{\infty}$ is only confirmed and have nothing to do with initial distribute. And when $1 \le i \le m$, $p_i'^{\infty} > 0$; when $\mathrm{m} \le \mathrm{i} \le \mathrm{n}$, $p_i'^{\infty} = 0$.

Theorem 1. In TSDE, for $\forall i,k \in \{1,2,...,|F|\}$, when $k \le i$, $p_{\mathrm{i,k}}>0$; when $k>i$, $p_{\mathrm{i,k}}=0$.

The method in similar literature [13] and [14] is easy to prove the above theorem come into existence. The theorem 1 show that in TSDE , the population with lower fitness value can transfer to the sameness or the population with higher fitness value, the inverse is not true. So, once the algorithm enter into group $\boldsymbol{P}_1$, it is impossible to escape from it.

Theorem 2. The TSDE algorithm is global convergence.

Proof: Each $\boldsymbol{P}_{\mathrm{i}}$ (i=1,2,…,$|F|$) can be seen to a state on limited Markov chain. According to Theorem 1, the transfer matrix of the above Markov chain express following:

$$\mathrm{P}' = \begin{bmatrix} p_{1,1} & 0 & \cdots & 0 \\ p_{2,1} & p_{2,2} & \cdots & 0 \\ \vdots & & & \vdots \\ p_{|F|,1} & p_{|F|,2} & \cdots & p_{|F|,|F|} \end{bmatrix} = \begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$$

And $R = (p_{2,1}, p_{3,1}, \cdots, p_{|F|,1})^T > 0$, $T\neq 0$, $C=(p_{1,1})=(1)\neq 0$. Thus, according to Theorem 1, we can draw the conclusion: $\mathrm{P}'^{\infty} = \lim_{n\to\infty} \mathrm{P}'^{k} = \lim_{n\to\infty} \begin{bmatrix} C^k & 0 \\ \sum_{i=0}^{k-1} T^i R C^{k-i} & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & 0 \\ R^{\infty} & 0 \end{bmatrix}$.

In the above equation, $C^{\infty} = 1$, $R^{\infty} = (1,1,\cdots,1)^T$. So, P'^{∞} is a steady random matrix, and $\mathrm{P}'^{\infty} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}$. In this way, according to Theorem 2, the TSDE algorithm is global convergence.

5 Experimentation Results and Analysis

In order to validate the feasibility and effectivity of TSDE, to the following 5 typical Benchmark functions, separately use TSDE, DE1, DE3, DE5 and DEfirDE[6] to emulatively compute, and then contrast the best solution, the worst solution, the mean and the standard deviation concluded from the thirty times independently running. The hardware environment used is DELL Pentium(R)4-CPU1.70 GHz, 128MB memory, and implement by the use of VC++ 6.0.

$$f_1(X)=0.5-\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{(1+0.001(x_1^2+x_2^2))^2},\ \text{and } X=(x_1, x_2),\ -10\le x_1, x_2\le 10.$$

$$f_2(X)=\sum_{i=1}^{5}[i\cos((i-1)x_1+i)]\sum_{j=1}^{5}[j\cos((j+1)x_2+j)]+(x_1+1.42513)^2+(x_2+0.80032)^2,$$

and $X=(x_1,x_2, x_3,x_4, x_5)$, $-10\le x_i\le 10$, i=1,2,3,4,5.

$$f_3(X)=\sum_{i=1}^{n-1}[100(x_{i+1}^2-x_i)^2+(1-x_i)^2],\ -2.048\le x_i\le 2.048,\ i=1,2,\ldots,n.$$

$$f_4(X)=\sum_{i=1}^{n}x_i^2,\ \text{and } X=(x_1,x_2,\ldots,x_n),\ -100\le x_i\le 100,\ i=1,2,\ldots,n.$$

$$f_5(X)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})+1,\ \text{and}\ \ X=(x_1,x_2,\ldots,x_n),\ -600\le x_i\le 600,$$

i=1,2,…,n.

In the use of TSDE, DE1, DE3 and DE5 to compute f_1 and f_2, the size of population is 20. When computing f_3~f_5 , the size of population is 50. The maximum iterative number of each algorithm generally is 3000, α=0.5, CR=0.3, and separately compute 30 times. The computation error is 10^{-20}, the result hold at least 3 valid figures. Table 1 and Table 2 present the best solution, the worst solution, the mean and the standard deviation concluded by each algorithm.

Table 1. The computation results of f_1~f_5 concluded by TSDE and DE1

Problem		TSDE				DE1			
F	*d*	*Best*	*Worst.*	*Mean*	*Std.Dev*	*Best*	*Worst.*	*Mean*	*Std.Dev*
f_1	2	1.00	1.00	1.00	0.0	1.00	1.00	1.00	0.0
f_2	2	-176.138	-176.138	-176.138	0.0	-176.138	-176.138	-176.138	0.0
f_3	30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_4	30	0.0	0.0	0.0	0.0	4.90e-12	2.45e-11	1.04e-11	6.55e-13
f_5	30	0.0	0.0	0.0	0.0	4.54e-11	1.27e-10	1.12e-10	2.01e-15

Table 2. The computation results of f_1~f_5 concluded by DE3 and DE5

Problem		DE3				DE5			
F	*d*	*Best*	*Worst.*	*Mean*	*Std.Dev*	*Best*	*Worst.*	*Mean*	*Std.Dev*
f_1	2	9.90e-10	9.90e-10	9.90e-10	0.0	1.00	9.98 e-1	9.99 e-1	1e-3
f_2	2	-176.138	-176.138	-176.138	0.0	-176.138	-176.138	-176.138	0.0
f_3	30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_4	30	0.0	0.0	0.0	0.0	0.0	6.15e-17	1.23e-17	2.46e-19
f_5	30	9.86e-3	9.86e-3	9.86e-3	0.0	0.0	0.0	0.0	0.0

In order to contrast the superior and inferior between TSDE and DEfirDE, to the high dimension function f_3~f_5 which are 100 dimensions, the size selected of population of each algorithm is 100. In TSDE, set α=0.5, CR=0.3, and the maximum iterative number is 50,000. The maximum iterative number of DEfirDE is 500,000, and the setting of other parameters refer to literature [6]. Table 3 present the mean and the standard deviation of best solution concluded by the two algorithms separately running 30 times.

Table 3. The computation results of f_3~f_5 concluded by TSDE and DEfirDE

Problem		TSDE		DEfirDE	
F	*d*	*Mean of Best*	*Std.Dev of Best*	*Mean of Best*	*Std.Dev of Best*
f_3	100	9.230016	11.02759	107.5604	28.2529
f_4	100	1.0 e-20	0.0	1e-6	0.0
f_5	100	4.76e-16	5.49e-12	1e-6	0.0

It can be seen from the computation results of table 1 to table 3, for the two dimensions function f_1 and f_2, the solution between TSDE, DE1,DE3 and DE5 is basic sameness. But for high dimension function f_3~f_5 , under the same parameter setting, the best solution, the mean and the standard deviation of the best solution solved by TSDE are better than DE1, DE3 and DE5. In the case of the maximum iterative number is lesser, the result solved by TSDE is better than solved by DEfirDE. So, it can come to the conclusion that the adaptability of TSDE is better than other algorithm, and for the high dimension optimizing function, with the gradually increase of function dimension, the superiority of global convergence and robustness is more and more great and the solution quality is improved distinctly. All the above show that TSDE is a feasible improved algorithm, which is adapt to solve the high dimension complex optimizing function.

6 Conclusion and Prospect

This paper first analyse the characteristic of DE, and then present an effective improved differential evolution algorithm—TSDE based on three effective evolutionary strategy, and use the random Markov chain theory to prove the global convergence. The imitation experimentation of 5 typical Benchmark function show that the global convergence, adaptability and robustness of TSDE is better than the original DE and DEfirDE algorithm, which shows that the method presented by the paper is not only feasible, but also effective. Due to DE is a relatively novel evolution algorithm, the problem existing is needed to be research further. For example, the research on the strong convergence and convergence speed of DE is the importance to the future research. Further more, according to dynamically distribute the proportion of each evolution strategy, whether the improved algorithm TSDE presented by the paper has the ability to improve the performance or not is needed to thoroughly analyze and discuss in the future.

References

1. Storn, R., Price, K.: Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
2. Price, K.: Differential evolution: A fast and simple numerical optimizer. In: Proceedings NAFIPS 1996, pp. 524–525 (1996)
3. Abbass, H.: Self-adaptive pareto differential evolution. In: Proceedings of the IEEE 2002 Congress on Evolutionary Computation, Honolulu, Hawaii, pp. 831–836. IEEE Press, Los Alamitos (2002)
4. Babu, B., Jehan, M.M.L.: Differential evolution for multiple-objective optimization. Evolutionary Computation 4, 8–12 (2003)
5. He, Y., Wang, X., Kou, Y.: A Binary Differential Evolution Algorithm with Hybrid Encoding. Journal of Computer Research and Development 44(9), 1476–1484 (2007)
6. Noman, N., lba, H.: Enhancing differential evolution Performance with Local Search for High Dimensional Function Optimization. In: GECCO 2005, Washington, DC, USA, pp. 967–974 (2005)
7. Zhao, G., Peng, X., Sun, N.: A modified differential evolution algorithm with hybird optimization strategy. Acta Electronica Sinica 34(12), 2402–2405 (2006)
8. Cruz, I.L.L., Van Willigenburg, L.G., Van Straten, G.: Efficient differential evolution algorithms fomulatimodal optimal control problem. Application Soft Computing 3, 97–122 (2003)
9. Babu, B.V., Munawar, S.A.: Different Evoluntion for the Optimal Desig of Heat Exchangers. In: Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development:A Vision 2010 and Beyond, IE(I) (2000)
10. Rajive, J., Sanderson, A.C.: Minimal representation multisensor fusion using differential evolution. In: Proceedings 1997 IEEE International symposium on Computational Intelligence in Robotics and Automation (1997)
11. Liu, Y., Kang, L., Chen, Y.: Nonnumerical parallel algorithms-genetic algorithm. Science Press, Beijing (2003)
12. Iosifescu, M.: Finite Markov processes and their applications. Wiley, Chichester (1980)
13. Dinabandhu, B., Murthy, C.A., Sankar, K.P.: Genetic algorithm with elitist model and its convergence. International Journal of Pattern Recognition and Artifical Intelligence 10(6), 731–747 (1996)
14. Jang, R., Luo, Y., Hu, D., Yip, S.K.: A genetic algorithm by coordinating exploration and exploitation–Convergence properties and performance analyses. Chinese Journal Computers 24(12), 1233–1241 (2001)

Cluster-Degree Analysis and Velocity Compensation Strategy of PSO

Quansheng Dou[1,2,3], Zhijun Yu [4], Zhongzhi Shi [1], Erkeng Yu[3], and Yongzhi Zheng[3]

[1] School of Computer Science and Technology, Shandong Institute of Business and Technology, Yantai, China
Li_dou@163.com
[2] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
shizz@ics.ict.ac.cn
[3] Yantai Dongfang Electronics Information Industry Group Co, Ltd, Yantai, China
erkyu@mx.cei.gov.cn, yongzhi.zheng@dongfang-china.com
[4] School of Nursing, Binzhou Medical University, Yantai, China
ekinkaiser@hotmail.com

Abstract. The particle's trajectory in particle swarm was fully analyzed in this paper, the theorem about necessary condition of convergence property of PSO was proposed by solve difference equation. Based on this theorem, discussed the influence of random parameters on particle's trajectory, the concept of cluster-degree was put forward and distribute status of particle with different cluster-degree was studied. The reasonable parameters setting range based on cluster-degree was proposed, at the same time, velocity compensation of particle's velocity was proposed in order to improving performance of PSO. So this paper is helpful for the choosing and adjustment of PSO parameters in practical application.

Keywords: particle swarm optimization, convergence property, cluster-degree.

1 Introduction

Particle swarm optimization (PSO) was an evolutionary computing technology designed by Kennedy and Eberhart in 1995[1][2] and has been applied successfully in various optimization problems[3]~[8]. In PSO every potential solution is a "particle" in the search space. Each particle may be expressed by (x_i, v_i, p_i), and the next position of the particle can be defined by formula 1

$$\begin{aligned} v_{t+1} &= wv_t + c_1 r_1 (p_i - x_t) + c_2 r_2 (p_g - x_t) \\ x_{t+1} &= x_t + v_{t+1} \end{aligned} . \tag{1}$$

where x_i is the current position of the particle, v_i is the current velocity of the particle , p_i is the best position passed by particle i, p_g is the best position passed by

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 98–106, 2008.
© Springer-Verlag Berlin Heidelberg 2008

all the particles in the swarm ; c_1、c_2 are positive constants, r_1、r_2 are random numbers between $[0,1]$, ω is the inertia parameter. There is lots of theoretical analyses about PSO, most of them suppose random numbers r_1、r_2 are their mathematical expectations, this sort of work is called certain analysis in this paper, the necessary condition of particle convergence is proposed by certain analysis, and the new conclusion consummates the previous research results. Based on this foundation, the concept of cluster-degree was put forward so as to show the distributing condition during the moving process of particles, and according to the cluster-degree, the concrete range of the particle parameters is given, which is helpful for the choosing and adjustment of PSO parameters in practical application.

2 Necessary Condition of Convergence Property of PSO

Now the analysis of convergence property of PSO mostly took advantage of certain analysis, and the mathematical expectations of r_1、r_2 are $\frac{1}{2}$, let $\varphi_1 = \frac{c_1}{2}$、$\varphi_2 = \frac{c_2}{2}$, $k = \varphi_1 + \varphi_2 = (c_1 + c_2)/2$. Under this hypothesis, [9][10] have done many significant jobs. The following theorem 1 gives the parameter range of particle astringency, which is the supplement of the previous research results.

Theorem 1. under the certain analysis condition, the parameter setting range in which particle's trajectory will converge is a triangle area surrounded by three lines $k = 2w + 2$、$k = 0$、$w = 1$.

Proof: Under the certain analysis condition, formula 1 may be rewritten to the following form:

$$x_{t+2} - \alpha x_{t+1} + w x_t - \beta = 0. \tag{2}$$

Where $\alpha = 1 + w - k$, $\beta = \varphi_1 p_i + \varphi_2 p_g$, $k = \varphi_1 + \varphi_2$ formula 2 is a two-order linear difference equation, its characteristics equation is:

$$\lambda^2 - \alpha\lambda + w = 0. \tag{3}$$

$\Delta = \alpha^2 - 4w = (1 + w - k)^2 - 4w$ is the discriminant of formula 3, there are three different cases according to Δ

Case 1: When $\Delta > 0$, the characteristics equation has two real roots, and the root of formula 2 has the following form:

$$x_t = C_1 \lambda_1^t + C_2 \lambda_2^t. \tag{4}$$

where C_1、C_2 are arbitrary constants, λ_1、λ_2 are two real roots of characteristics equation 3. Conclude from formula 4: when both of the absolute values of λ_1、λ_2 are

smaller than 1, x_t will converge. Obviously when $w<0$ or $w=0$ and $k\neq 1$, $\Delta>0$; and when $w>0$

$$\begin{aligned}\Delta &= (1+w-k)^2-4w \\ &= (1+w-k+2\sqrt{w})(1+w-k-2\sqrt{w})\end{aligned}.$$

So when $k>1+w+2\sqrt{w}$ or $k<1+w-2\sqrt{w}$, $\Delta>0$; and now:

$$\lambda_{1,2}=\frac{\alpha\pm\sqrt{\Delta}}{2}=\frac{\alpha\pm\sqrt{\alpha^2-4w}}{2}. \tag{5}$$

Let $\lambda_1=\frac{\alpha-\sqrt{\alpha^2-4w}}{2}$, $\lambda_2=\frac{\alpha+\sqrt{\alpha^2-4w}}{2}$, when $\lambda_1>-1$ and $\lambda_2<1$, both of the absolute values of λ_1、λ_2 are smaller than 1, and it is easy to prove that if $\lambda_1>-1$, then $k<w+3$ and $k<2w+2$; and if $\lambda_2<1$, then $k>0$ and $k>w-1$; so the convergence condition when $\Delta>0$ can be concluded.

Case 2: When $\Delta=0$, characteristics equation 3 has only one real root, and the root of formula 2 has the following form:

$$x_t=C_1\lambda^t+C_2t\lambda^t. \tag{6}$$

Where C_1、C_2 are arbitrary constants, λ is the real root of characteristics equation 3. When $|\lambda|<1$, x_t will converge.

Study the discriminant Δ of formula 3, it is easy to prove that if $w=0$ and $k=1$, then $\Delta=0$; and if $k=1+w+2\sqrt{w}$ or $k=1+w-2\sqrt{w}$, then $\Delta=0$; and now:

$$\lambda=\frac{\alpha}{2}=\frac{1+w-k}{2}. \tag{7}$$

It is easy to prove that if $|\lambda|<1$, then $k>w-1$ and $k<w+3$; so the convergence condition when $\Delta=0$ can be concluded.

Case 3: When $\Delta<0$, characteristics equation 3 has a pair of conjugated complex roots, and the root of formula 2 has the following form:

$$x_t=C_1R^t\cos t\theta+C_2R^t\sin t\theta. \tag{8}$$

Where C_1、C_2 are arbitrary constants, $R=\sqrt{w}$, when $R<1$, x_t will converge. Study the discriminant Δ of formula 3, it is easy to prove that when

$w>0$, $k<1+w+2\sqrt{w}$ or $k>1+w-2\sqrt{w}$, $\Delta<0$; and $w<1$, the differential equation 2 has convergence solution.

Synthesized from the above three cases, the theorem was proved. The convergence condition in the theorem1 is shown as figure 1:

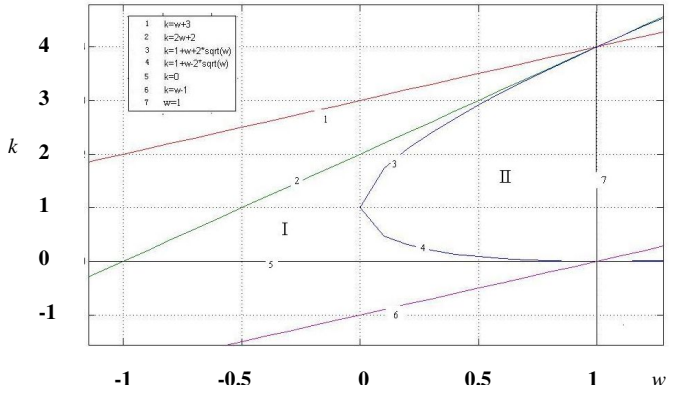


Fig. 1. The interzone surrounded by line 2, 5, 7 is the parameter range in which particle will converge. line 1:k=w+3; line 2:k=2w+2; line 3: k=1+w+2w$^{1/2}$;line 4:1+w-2w$^{1/2}$; line 5: k=0; line 6: k=w-1; line 7: w=1.

In fact, the conclusion in [10] is the part of the triangle area when $w>0$, most of the research ignored the condition when $w<0$. There are appropriate parameters which can astringe the particle running trajectory when $w<0$. In figure 1 the triangle area is composed of two parts, which respectively correspond to two different conditions $\Delta>0$ (area I) and $\Delta\le 0$ (area II), although parameters in two interzone can all let the particles converge, but the way of converge is different, which is related to the different root form under different cases.

3 The Cluster-Degree of Particle and the Convergence Analysis Based on the Cluster-Degree

Let $x_{i0}, x_{i1}, x_{i2}, \ldots x_{iT}$ represent the all the positions passed by particle x_i with parameters w, c_1, c_2, define the cluster-degree about w, c_1, c_2 is

$$\kappa = N / TP . \tag{9}$$

Where $N = num(x_j \in (L,H))\ \ j = 0,1...,T$, $L = \min(p_i, p_g)$, $H = \max(p_i, p_g)$, is the number of particle fall between p_i, p_g , $P = \frac{1}{\sigma\sqrt{2\pi}} \int_{u-\sigma}^{u+\sigma} e^{\frac{-(t-u)^2}{2\sigma^2}} dt$, $u = (p_i + p_g)/2$, $\sigma = \frac{|p_g - p_i|}{2}$.

Experiments show that the cluster-degree has nothing to do with p_i and p_g, so the reasonable range of w, c_1, c_2 can be defined by cluster-degree. Normally, what kind of cluster-degree can get the best solution, it is different for different problems, but one point is affirmative: if the particle cluster-degree is too large, lots of the particles will distribute at the border of the searching border, and cannot get the effective search; if the particle cluster-degree is too small, then the particles will converge between the swarm optimum and the individual optimum, premature phenomenon will take place. Figure 2 gives two curves when κ=0.5 and κ=1. Normally, the parameter setting should keep the cluster-degree within this range.

The cluster-degree reflects the distribution of the particles in the searching process, only by reasonable parameter setting which keeps the cluster-degree of particle not too large and not too small, PSO may find the best solution. Here the definition of the cluster-degree is based on the situation which suppose the global optimum and individual optimum is fixed, although p_g and p_i are fixed, unreasonable cluster-degree will make the particle distribute all over the search space or congregate in small area of search space, so when p_g and p_i changed frequently, it is impossible that particle with the parameters with unreasonable cluster-degree can get ideal solution.

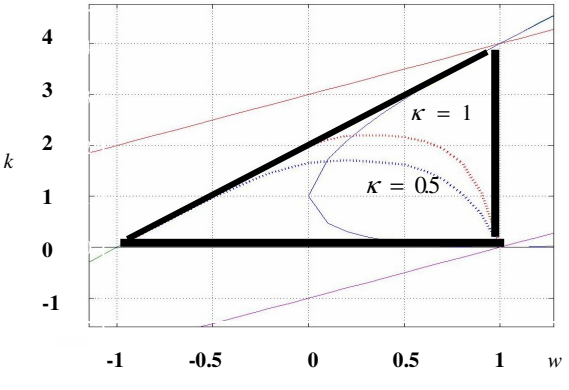


Fig. 2. Reasonable cluster-degree range, value of cluster-degree between 0.5 and 1

Because of the dynamic change of p_g and p_i, not only need to consider whether the cluster-degree is reasonable, but also need to take it into account whether the convergent velocity of particle is reasonable. In PSO, the particles affect each other by

the global optimum p_g, the distance of global optimum p_g and the individual p_i will become short rapidly if the velocity of particles converges too fast. According to the analysis above, we know that the particle take a normal distribution with $\mu = \frac{c_1 p_i + c_2 p_g}{c_1 + c_2}$ as its center and $\sigma = \alpha(\frac{|p_g - p_i|}{2})$ as its variance. The distance between p_g and p_i is short, which will then diminish the particle's velocity *v*, even there is a reasonable cluster-degree, the fast attenuation of particle's velocity *v* will make p_g and p_i approach each other rapidly, which will form prematurity phenomenon.

By the analysis above, we know that to make PSO get ideal solution, there are two conditions need to meet: (1) keep the particle with reasonable cluster-degree (2) the particle velocity *v* can't attenuate too fast while keeping the reasonable cluster-degree. When the particle parameter setting meets area II in figure 2, by formula 8 can conclude that, the attenuation velocity of particle is determined by $\sqrt{w}$, and $0 < w < 1$, when w is large, the attenuation of particle velocity is also slow. Whereas if *w* is small, the particle velocity attenuates more rapidly, it is easy to form prematurity.

The following experiments proves the above analysis, 23 benchmark function described in [11] were adopted in our experiments; table 1 lists the average result of 10 experiments under different cluster-degree conditions of four functions. In order to validate the effect of velocity attenuation with the same cluster-degree, under the same group of parameters, velocity compensation strategy is used, i.e. when the iteration times and the particle velocity respectively less than a threshold, assign a new velocity to the particle so as to make up the rapid attenuation of velocity.

The first group of parameters w=0.729 c_1=1.49445 c_2=1.49445, lots of experiments have proved that these parameters have better searching effect, and its cluster-degree $\kappa = 0.8021$. The cluster-degrees from the second to the sixth group of parameters are close to the cluster-degree of the first group (cluster-degree is a statistical value, with the same group of parameters, every computed cluster-degree is close but not the same), the *w* values of second and third group of parameters are respectively 0.8 and 0.9. According to the above analysis, when the parameters are set in interzone II described in theorem 1, from formula 8 we know that the velocity of particle convergence is determined by $\sqrt{w}$, with these two parameter settings, the particle velocity won't attenuate rapidly, and their cluster-degrees are close to the first

Table 1. With different cluster-degree, normal running and velocity compensation strategy, the average result of 10 experiments. Dimensions of four benchmark function in this table are 30 and max iterative times are 2000.

No.	Paras	Velocity Compensation	f_1	f_5	f_9	f_{11}
1	w=0.729 c_1=1.49445 c_2=1.49445 κ =0.8021	No	<1e-8	32.23	<1e-8	<1e-8
		Yes	<1e-8	21.44	<1e-8	<1e-8
2	w=0.8 c_1=1.2784 c_2=1.2784 κ =0.8019	No	<1e-8	30.24	<1e-8	<1e-8
		Yes	<1e-8	20.98	<1e-8	<1e-8
3	w=0.9 c_1=0.7942 c_2=0.7942 κ =0.8016	No	<1e-8	34.85	<1e-8	<1e-8
		Yes	<1e-8	21.72	<1e-8	<1e-8
4	w=0.1 c_1=1.8568 c_2=1.8568 κ =0.8017	No	295.37	2069.63	44.635	125.52
		Yes	<1e-8	33.879	3.5e-4	<1e-5
5	w=0.2 c_1=1.8711 c_2=1.8711 κ =0.8022	No	141.69	1980.23	33.84	119.37
		Yes	<1e-8	31.56	2.7e-4	<1e-5
6	w=0.2 c_1=2.1851 c_2=2.1851 κ =0.5	No	2564.35	24536.29	452.35	856.21
		Yes	2487.33	20336.76	266.87	795.54
7	w=0.729 c_1=1.2995 c_2=1.2995 κ =1.2	No	254.13	984.65	86.35	231.42
		Yes	23.14	206.89	46.25	10.21

group. From the experimental data find that these two groups of parameters get ideal searching results, which indicates that under the premise of reasonable particle velocity v attenuation, same cluster-degrees have close searching results. Under the direction of this analysis result, we found many parameters in the interzone $w > 0.729$ which can get better searching results, here, we don't enumerate them all.

For the fourth and fifth group of parameters, the cluster-degree is close to the first group, but the w value is relatively small, according to formula 8, the particle's velocity attenuates more rapidly, so the searching result is not satisfied. Take advantage of the following compensation strategy: when the iteration times $t < \frac{2}{3}T$, and

$\sum_{i=1}^{dims} | v_i | < 1e-4$, reset v_i, let $v_i = rand(0,1)$, here T is the maximum of the iteration times, and v_i is the *ith* component of the particle's velocity v. With these two groups of parameter setting, using velocity compensation strategy, get better solutions too. But the searching efficiency is slightly low.

The cluster-degree of the sixth group of parameters is 0.5, the searching result is bad in normal condition, and the result has not any essential improvement with the velocity compensation strategy, which means that when the cluster-degree is small, the particles distribute dispersedly in the searching space and can't search effectively, the velocity compensation can't solve the problem of particle dispersing. So the searching result can't be improved.

The cluster-degree of the seventh group of parameters is 1.2, the searching result isn't satisfactory, after taking advantage of velocity compensation strategy, searching result is improved slightly, which means that the particle with parameters with large cluster-degree can not get satisfactory solutions.

Setting the particle parameters to keep suitable cluster-degree is the necessary condition to get ideal searching result, and under the premise that the cluster-degree is reasonable, the attenuation velocity of particle velocity *v* should also in a reasonable range, when the attenuation velocities of *v* are close, the searching results of close cluster-degree are also close.

4 Conclusion

This paper gives a more complete analysis of PSO. The necessary condition of the particle convergence was given under certain condition, which complements the previous research. Based on it, proposes the cluster-degree concept, the distribution status was studied by this concept, and gives the reasonable cluster-degree range and more concrete parameter setting interzone. At the same time, propose the particle velocity compensation strategy during the running process. The strategy can improve the searching performance of PSO when the cluster-degree is a bit large or reasonable but the velocity *v* attenuates too fast. It is helpful for the choosing and adjustment of PSO parameters in practical application.

Acknowledgments. This work was supported by the China National Nature Science Fund (grants 60773053), Postdoctor Research Fund of Shandong Province (grants 20070311), Plan Project of Education Department of Shandong Province (grants J06G04), Project of Science and Technology Bureau of Yantai City (grants 2006150).

References

1. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
2. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948. IEEE Service Center, Piscataway
3. Clerc, M.: TRIBES-Aparameter Free Particle Swarm Optimizer, http://clerc.maurice.free.fr/PSO2002-08-10/2003-10-08
4. Salman, A.: Discrete Particle Swarm Optimization for Heterogeneous Task Assignment Problem. In: Proceedings of World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) (2001)
5. Clerc, M.: Discrete Particle Swarm Optimization: A Fuzzy Combinatorial Black Box, http://clerc.maurice.free.fr/PSO/Fuzzy_Discrete_PSO/Fuzzy_DPSO.htm.2000-04-01/2003-10-08
6. Hirotaka, Yoshida, Kenichi: A particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Stability. In: IEEE International Conference on Intelligent System Applications to Power Systems, Rio de Janeiro (1999)
7. Voss, M.S., Feng, X.: Arma Model Selection Using Particle Swarm Optimization and Aic Criteria. In: 15th Triennial World Congress, Barcelona, Spain (2002)
8. Clerc, M.: Some math about Particle Swarm Optimization, http://clerc.maurice.free.fr/pso/2002-08-10/2003-10-08
9. Van den Bergh, F.: An analysis of Particle Swarm Optimizers: [Ph D dissertation]. University of Pretoria, Pretoria (2001)
10. Li, N., Sun, D.-b., Zou, T., Qin, Y.: An Analysis for a Particle's Trajectory of PSO Based on Difference Equation. Chinese Journal of Computers 29(11) (November 2006)
11. Dou, Q.-s.: Research on Evolutionary Computing for Optimization Problem. [Ph D dissertation]. Jilin University, Changchun (2005)

Combine LHS with MOEA to Optimize Complex Pareto Set MOPs

Jinhua Zheng, Biao Luo, Miqing Li, and Jing Li

Research Center of Evolutionary Computation and Intelligent System, Xiangtan University, 411105, Hunan, China
jhzheng@xtu.edu.cn, biao.luo@hotmail.com, limit1008@126.com, laoer88888@163.com

Abstract. The Pareto set (PS) of real multi-objective optimization problems (MOPs) are often unknown and complex, so, it is significant for multi-objective evolutionary algorithms (MOEAs) to solve complex PS MOPs (CPS_MOPs namely). In this paper, we combined Latin hypercube sampling (LHS) with MOEA, proposed a LHS based MOEA (LHS-MOEA). We suggested two kinds of LHS-MOEA, in which LHS local search and evolutionary operator are combined to handle CPS_MOPs. Through some experiments, the results demonstrate that LHS-MOEA performs much better than the traditional prevalent MOEA — NSGA-II in solving CPS_MOPs.

Keywords: Multi-objective Evolutionary Algorithm, Latin Hypercube Sampling, Complex Pareto Set, Local Search, Evolutionary Operator.

1 Introduction

Till now, most studies about multi-objective evolutionary algorithms (MOEAs) have been focused on objective space (OS), little attention concerned on decision space (DS). But in real cases, Pareto set (PS) in DS are often unknown and complex. Therefore, it is significant for MOEAs to solve complex PS multi-objective optimization problems (CPS_MOPs). Traditional MOEAs, such as, NSGA-II [1], SPEA-II [2], PAEA [3] et al, are effective on solve simple PS MOPs, but some studies [4] found that it is hard for traditional MOEAs to solve CPS_MOPs. Because there is serious degradation when use traditional MOEAs to solve CPS_MOPs. In order to prevent the degradation, we combined a famous sampling method — Latin hypercube sampling (LHS) with MOEA, proposed a LHS based MOEA (LHS-MOEA), in which LHS local search and evolutionary operators are suggested. Individual based LHS-MOEA (LHS-MOEA/I) and population based LHS-MOEA (LHS-MOEA/P) are realized. Compared with the traditional prevalent MOEA — NSGA-II, the experimental results demonstrate that LHS-MOEA is an effective algorithm in solving CPS_MOPs.

The rest of this paper is organized as follows. In section 2, some issues about CPS_MOPs are discussed. Two kinds of LHS-MOEAs are realized in section 3. In section 4, CPS_MOPs obtained by linear variable linkages and nonlinear variable linkages are tested with NSGA-II, LHS-MOEA/I and LHS-MOEA/P. The experimental results are briefly analyzed. Section 5 gives the conclusions.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 107–116, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Some Issues of CPS_MOPs

CPS_MOP is a special kind of MOP. For the PS of CPS_MOP is more complex, it is more difficult for MOEAs. In this section, some issues about CPS_MOPs are discussed.

2.1 Problem Description

A universal definition of MOPs can be given as follow:

$$\text{Min } \vec{f}(X) = (f_1(X), f_2(X), \cdots, f_r(X)) \tag{1}$$

$$g_i(X) \geq 0; (i = 1, 2, ..., k) \tag{2}$$

$$h_i(X) = 0; (i = 1, 2, ..., l) \tag{3}$$

where $\vec{f}(X)$ is the objective vector, r is the dimension of objectives, (2) and (3) are equality-constraints and inequality-constraints, $X = (x_1, x_2, ..., x_n)$ is variable vector, n is the dimension of variables, $X \in \Omega$, $\Omega \subseteq R^n$,where Ω is the feasible space, then, $\vec{f} : \Omega \to \Pi$, $\Pi \subseteq R^r$, Π is the objective space.

The task of optimization of MOPs is to find Pareto solutions $X^* = (x_1^*, x_2^*, ..., x_n^*)$. In MOEAs, Pareto solutions of current population are often called non-dominated solutions and the Pareto solutions set is often called non-dominated set (NDS). Curve of NDS painted in the objective is named as Pareto front (PF). PF_{True} is the true Pareto front of MOP.

As one special kind of MOPs, CPS_MOPs have all characters of MOPs. But the PS of CPS_MOPs is more complex. In this paper, CPS_MOPs can be obtained by variable linkages (variable linkages will be discussed in section 2.3).

2.2 Degradation in Traditional MOEAs

Most traditional MOEAs are effective in solving MOPs with simple PS, such as: ZDT series [1, 5-7] (e.g. PS of ZDT1: all variables equal to zero except the first variable), but a study [4] found that it is hard for traditional MOEAs to solve CPS_MOPs. When use traditional MOEAs to solve CPS_MOPs, degradation will produced (Fig. 1 and Fig. 2). As shown in figures, the population shown in Fig. 1 is close to PS, but the next population in Fig. 2 is far away from PS again in traditional MOEAs.

2.3 Variable Linkages

CPS_MOPs are obtained by variable linkages [7] based on ZDT series. Variable linkages used in this paper are linear variable linkages and nonlinear variable linkages which can be described as follows. Some detailed information about this issue, one is refer to [7,8].

(1) Linear variable linkages
Let x_1 invariable, replace x_i (i=2,…,n) by $x_i - x_1$.

(2) Nonlinear variable linkages
Let x_1 invariable, replace x_i (i=2,…,n) by $x_i^2 - x_1$.

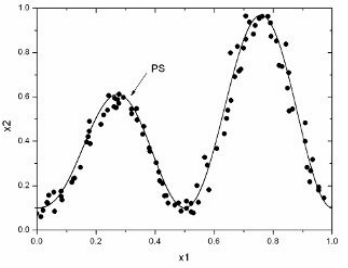


Fig. 1. Shows population is close to PS

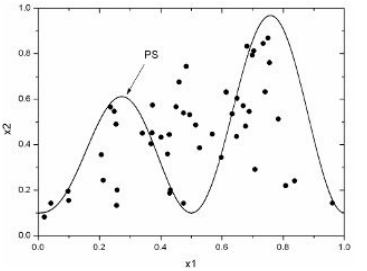


Fig. 2. Shows next population obtained by evolutionary operator

3 LHS Based MOEA(LHS-MOEA)

As discussed in above paragraphs, we realized that to prevent degradation is the key point to solve CPS_MOPs. In this paper, we combined LHS with MOEA, proposed a LHS based MOEA (LHS-MOEA), in which LHS local search and evolutionary operator are combined to generate next population.

3.1 LHS Local Search

LHS first proposed by Mckay et al [9]. It is a hierarchical sampling method which has been applied widely. Suppose a hypercube: $x^i \in [x_l^i, x_u^i]; (i = 1, 2, ..., n)$, sample size is H, n is the dimension of variables. Algorithm 1 shows the process of LHS.

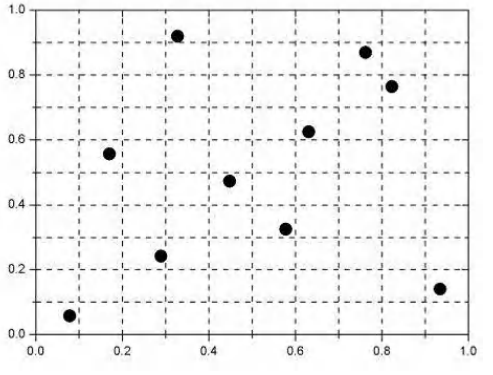


Fig. 3. Example of LHS

Algorithm 1: LHS

Step 1. Divide every interval $[x_l^i, x_u^i]$ into exactly H equal intervals. Then, the hypercube is divided into H^n small hypercubes.

Step 2. Generate a matrix $A \in I^{H \times n}$, I is an aggregate of all integers. Every row is a random arrangement of [1,2,…, H]. Then, every line of matrix A is a small hypercube which is be selected. Follow this rule; we can select H hypercubes.

Step 3. Generate a sample randomly in every selected hypercube, and then, H samples are generated.

Fig.3 shows an example of LHS, in which: $n = 2$; $H = 10$; and $A = \begin{pmatrix} 5 & 2 & 7 & 1 & 9 & 4 & 10 & 6 & 8 & 3 \\ 5 & 6 & 7 & 1 & 8 & 10 & 2 & 4 & 9 & 3 \end{pmatrix}^T$.

An important characteristic of LHS is that one and only one sample can be generated in every line and row. Obviously, LHS can obtain a set of uniformly distributed samples. Some researchers call LHS as a "space filling design". In this paper, LHS is used for local search for LHS can exploit a neighborhood effectively. For an individual X in the population, LHS local search is executed in individual's δ-neighborhood. Algorithm 2 gives the procedure of LHS local search. Individual's δ-neighborhood B_δ is a hypercube which can be defined as follows:

(Individual's δ-neighborhood): Suppose X is an individual in the population. Then, the δ-neighborhood: $B_\delta = \{Z \mid z_i \in [x_i - \delta_i, x_i + \delta_i]\}$

where $X = (x_1, x_2, ..., x_n)$,δ is the neighborhood radius, $Z = (z_1, z_2, ..., z_n)$, $\delta = (\delta_1, \delta_2, ..., \delta_n)$. So, B_δ is a hypercube with length of sizes equal to 2δ and the central of B_δ is X.

Algorithm 2: LHS local search

LHS local search is a procedure which use algorithm 1 generate H sample individuals in δ-neighborhood of individual X.

Algorithm 3: LHS-MOEA/I

```
Begin:
 t = 1;
  Initialize; set evolutionary generation (G), population size (N), neighborhood radius (δ), proportional coefficient (p), Crossover probability (Pc), Mutation probability (Pm), Sample size(H);
  Initialize population P(t); mergePop = Φ;
  Rank_Pop(Pop(t));
  While (t <=G)
  {
     k = 1;
      While (k <= p*N)
      {// this while loop is for LHS local search
                Select an individual X from P(t) with tournament selection;
                If(X.rank == 1)
                        set neighborhood radius of X as δ;
                Else
                        set neighborhood radius of X as 2δ;
            Use LHS local search (algorithm 2) generate H sample individuals in the neighborhood of X, and merge those individuals into mergePop;
                k = k+1;
      }// end while(k <= p*N)
      Use evolutionary operator generate (1-p)*N individuals based on P(t), and merge those individuals into mergePop;
      mergePop = mergePop ∪ Pop(t);                    // elitism is realized here
      Rank_Pop (mergePop);
      Pop (t+1) = New_Pop (mergePop);
      mergePop =Φ;
      t = t+1;
  }// end while (t <=G)
End.
```

3.2 Individual Based LHS-MOEA

In this paper, we designed LHS-MOEA/I and LHS-MOEA/P. In the process of LHS-MOEA/I, one part of next population (proportion is p) are generated by LHS local search and another part (proportion is 1-p) are generated by evolutionary operator. Here, evolutionary operator includes: tournament selection, simulated binary crossover [10, 11] (SBX) and polynomial mutation [12]. Algorithm 3 shows the procedure of LHS-MOEA/I.

In algorithm 3, the size of *P(t)* is N, the size of *mergePop* can be calculated as follows:

$$\begin{aligned} |\text{Merge_Pop}| &= p*N*H+(1-p)*N+N \\ &= (\text{H}-1)*p*N+2N \end{aligned} \tag{4}$$

Two important operators of MOEAs are described in algorithm 4 and algorithm 5.

Algorithm 4: Rank_Pop(*P*)

Step 1. *i*=1;
Step 2. Construct *NDS* for sub-population of *P*, in which all individuals' rank has not been assigned. Set non-dominated solutions' rank as *i* in the sub-population;
Step 3. If all individuals' rank in population *P* has been assigned, end algorithm 4. Else, keep on;
Step 4. *i* = *i*+1; got to Step2;

Algorithm 4 gives a method for finding NDS and ranking a population. Algorithm 5 shows a procedure for construct a new population from an old population. Here, new population is *P(t+1)*, old population is *mergePop*.

Algorithm 5: New_Pop (*mergePop*)

Step 1. *i* = 1; *P(t+1)* =*Φ*;
Step 2. Suppose N_i is the number of individuals in population *mergePop* whose rank equal to *i*, then, (1) If $|Pop(t+1)|+ N_i < N$, copy those N_i individuals into population *P(t+1)*, go to Step3; (2) If $|Pop(t+1)|+ N_i >= N$, sort those N_i individuals based on crowding distance [1]. Copy $N-|Pop(t+1)|$ individuals which have the larger crowding distance into population *P(t+1)*, return *P(t+1)*, end algorithm 5;
Step 3. *i* = *i+1*; goto Step2;

3.3 Population Based LHS-MOEA

In LHS-MOEA/P, LHS local search and evolutionary operator are used to generate next population alternately. Algorithm 6 shows the procedure of LHS-MOEA/P.

In algorithm 6, the size of *P(t)* is *N*, the size of *mergePop* can be calculated as follows:

$$|\text{Merge_Pop}| = N*H+N = (\text{H}+1)*N \tag{5}$$

3.4 Analysis for LHS-MOEA

In algorithm 3 and algorithm 6, if a selected individual for LHS local search is a non-dominated solution, set the neighborhood radius as δ, else, set the neighborhood radius as 2δ. In this way, LHS-MOEA can search more excellent individuals in smaller

neighborhood of non-dominated solution so that the convergence can be improved. And LHS-MOEA can generate individuals with diversity in a larger neighborhood of dominated solution so as to prevent immature convergence.

```
Algorithm 6: LHS-MOEA/P
Begin:
  t = 1;
   Initialize; set evolutionary generation (G), population size (N), neighborhood radius (δ), Cross-
       over probability (Pc), Mutation probability (Pm), Sample size(H);
   Initialize population P(t); mergePop = Φ;
   Rank_Pop(Pop(t));
   flag = 0;        //"0" for evolutionary operator, "1" for LHS local search
   While (t <=G)
   {
      If (flag == 0)
      {//use evolutionary operator generate next population
            Use evolutionary operator generate N individuals based on P(t), and merge those
                individuals into mergePop;
              flag = 1;
       }
       Else If (flag == 1)
      {// use LHS local search generate next population
              k = 1;
            While (k <= N)
              {// this while loop is for LHS local search
                    Select an individual X from P(t) with tournament selection;
                    If(X.rank == 1)
                          set neighborhood radius of X as δ;
                    Else
                          set neighborhood radius of X as 2δ;
                  Use LHS local search (algorithm 2) generate H sample individuals in the
                      neighborhood of X, and merge those individuals into mergePop;
                      k = k+1;
            }// end while(k <= N)
              flag = 0;
       }
       mergePop = mergePop ∪ Pop(t);                    // elitism is realized here
       Rank_Pop (mergePop);
       Pop (t+1) = New_Pop (mergePop);
       mergePop =Φ;
       t = t+1;
   }// end while (t <=G)
End.
```

There are some obvious discrepancies between LHS-MOEA/I and LHS-MOEA/P. But a common character in two LHS-MOEAs is the combination of LHS local search and evolutionary operator, which is the key difference between LHS-MOEA and traditional MOEAs.

LHS-MOEA can prevent the degradation in solving CPS_MOPs for two reasons: (1) LHS local search assure that a population which is close to PS do not apart from PS again and it can also exploit promising areas. This operator can prevent degradation and accelerate convergence. (2) Evolutionary operator can explore new areas in search space. This operator can maintain population's diversity so as to prevent immature convergence.

4 Experiments and Results

In this section, 4 CPS_MOPs are tested with three MOEAs: NSGA-II, LHS-MOEA/I and LHS-MOEA/P. NSGA-II is one of the most famous and prevalent traditional

Table 1. Parameter settings

Evolutionary generation (G)	Population size N(2D;3D)	*Neighborhood radius* δ_i *(i=1,…,n)*	*Proportional coefficient(p)*
500	200;300	0.05	0.5
Crossover probability (P_c)	Mutation probability(P_m)	*Sample size(H)*	
0.9	1/n	2n	

MOEA. The parameters settings of experiments for three MOEAs are listed in table 1. Italics in the table are only for LHS-MOEAs.

A larger evolutionary generation (*G*) and population size (*N*) are used so that three MOEAs can evolve fully. *2D* and *3D* denote 2-ojective and 3-objective. *n* is the dimension of variable vector.

4.1 Performance Metric

Here, we introduce a new performance metric: Inverted Generational Distance (IGD): this metric is designed for both convergence and diversity. IGD is shown in (6).

$$IGD = \frac{\sum_{v \in P*} d(v,P)}{|P*|} \tag{6}$$

where P* is a set of uniformly distributed points in PF_{True}, P is the NDS obtained by MOEAs, d(v,P) is the minimum Euclidean distance between *v* and the points in P. If |P*| is large enough to represent PF_{True} very well, IGD could measure both the diversity and convergence of P in a sense. To have a low value of IGD, P must be very close to the PF_{True} and cannot miss any part of PF_{True}. In our experiments, we set |P*| as 500. IGD has been recently used by some researchers [4,13], as its name suggests, it is an inverted variation of the widely-used generational distance (GD) performance metric [14]. The GD can only represent the average distance of the points in NDS to the PF_{True}, but cannot effectively measure the diversity of the NDS.

Table 2. Test problems of CPS_MOPs

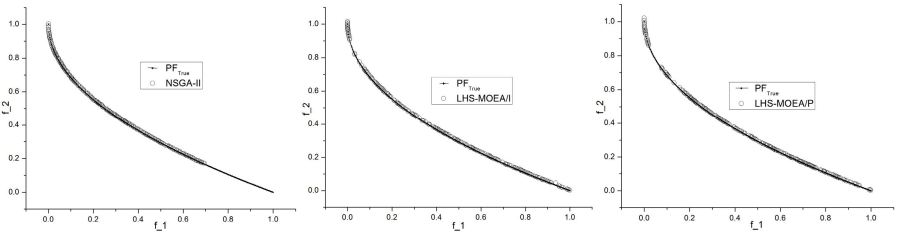
CPS _MOPs	Objective functions	Characters
F1	$f_1(x_1) = x_1;\ f_2(X) = g(1-\sqrt{(f_1/g)})$ $g(X) = 1+9\sum_{i=2}^{n}(x_i - x_1)^2/(n-1)$	Convex, n=20; $X \in [0,1]^n$, Linear variable linkages
F2	$f_1(x_1) = x_1;\ f_2(X) = g(1-(f_1/g)^2)$ $g(X) = 1+9\sum_{i=2}^{n}(x_i - x_1)^2/(n-1)$	Concave, n=20; $X \in [0,1]^n$, Linear variable linkages
F3	$f_1(X), f_2(X)$ *are the same to F*1; $g(X) = 1+9\sum_{i=2}^{n}(x_i^2 - x_1)^2/(n-1)$	Convex, n=20; $X \in [0,1]^n$, Nonlinear variable linkages
F4	$f_1(X), f_2(X)$ *are the same to F*2; $g(X) = 1+9\sum_{i=2}^{n}(x_i^2 - x_1)^2/(n-1)$	n=20; $X \in [0,1]^n$, Nonlinear variable linkages

4.2 Test Problems

Test problems of CPS_MOPs are listed in table 2. Those CPS_MOPs are obtained by variable linkages based on ZDT1 and ZDT2.

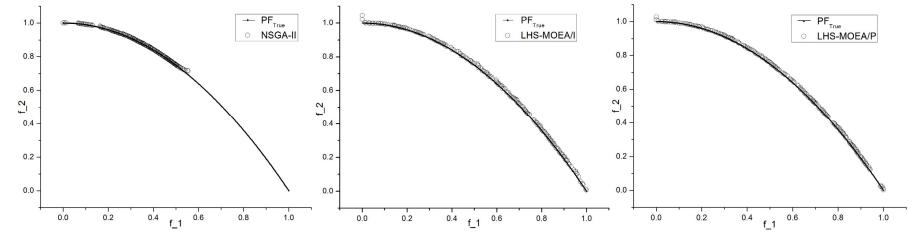
4.3 Experiments Results and Analysis

Fig.4-Fig.7 shows PFs of F1-F4 obtained by NSGA-II, LHS-MOEA/I and LHS-MOEA/P. For all CPS_MOPs test problems (F-F4), NSGA-II can only find a small part of PF_{True}, a majority of PF_{True} are missed. Especially, for F4, all non-dominated solutions obtained by NSGA-II are crowded in a small part of PF_{True}. However, both LHS-MOEA/I and LHS-MOEA/P obtain PFs which can almost cover with the whole PF_{True} and can obtain good convergence. In additional, NSGA-II can find more non-dominated solutions in PF_{True} on CPS_MOPs (F1,F2) based on linear variable linkages than CPS_MOPs (F3,F4) based on nonlinear variable linkages. It means that nonlinear variable linkages make PS more complex and traditional MOEAs are becoming less effective with the PS becoming more complex.



(a) NSGA-II (b)LHS-MOEA/I (c) LHS-MOEA/P

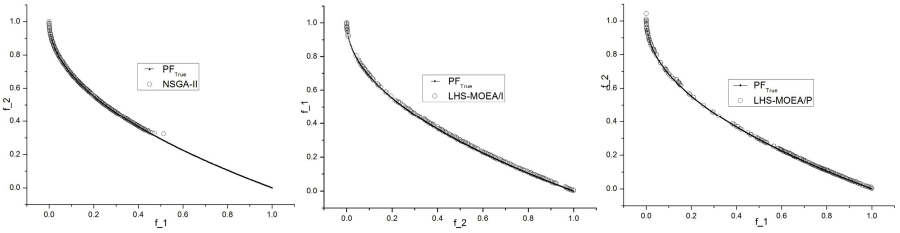
Fig. 4. PF of F1 use three MOEAs



(a) NSGA-II (b)LHS-MOEA/I (c) LHS-MOEA/P

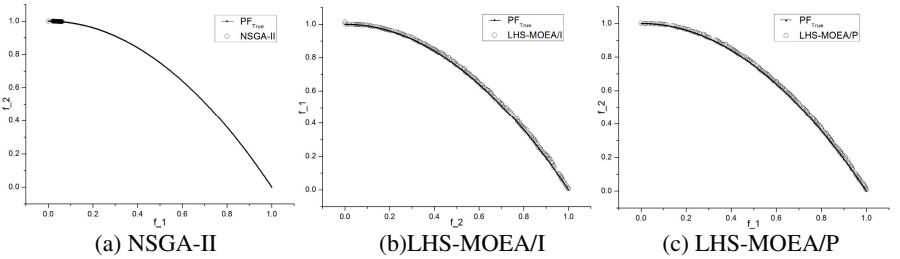
Fig. 5. PF of F2 use three MOEAs

A qualitative analysis for three MOEAs is given in above paragraphs. In order to give a quantitative analysis, we run NSGA-II, LHS-MOEA/I and LHS-MOEA/P 20 times on F1-F4. Mean IGD are listed in table 3. Bold numbers in the table are minimums. Apparently, mean IGD obtained by LHS-MOEA/I, LHS-MOEA/P are much less than NSGA-II, and mean IGD obtained by LHS-MOEA/I almost the same as



(a) NSGA-II (b)LHS-MOEA/I (c) LHS-MOEA/P

Fig. 6. PF of F3 use three MOEAs



(a) NSGA-II (b)LHS-MOEA/I (c) LHS-MOEA/P

Fig. 7. PF of F4 use three MOEAs

LHS-MOEA/P. In a word, experimental results show that LHS-MOEAs is much more effective than NSGA-II in solving CPS_MOPs.

Table 3. Mean IGD (20 independent runs of NSGA-II, LHS-MOEA/I and LHS-MOEA/P)

	F1	F2	F4	F5
NSGA-II	*0.097694*	*0.197088*	*0.245640*	*0.565686*
LHS-MOEA/I	**0.006012**	0.005956	**0.006704**	**0.006251**
LHS-MOEA/P	0.006685	**0.005923**	0.008765	0.246300

LHS-MOEA performs much better than traditional MOEA in CPS_MOPs, but any algorithm is impossible to excel other algorithms in all aspects according to David's *No Free Lunch theorems* (NFL) [15]. There is no exception for LHS-MOEA. LHS-MOEA needs sampling in individuals' neighborhood and those sampled individuals need fitness evaluations, which will increase time complexity accordingly.

5 Some Conclusions

In this paper, we proposed a LHS based MOEA (LHS-MOEA), in which LHS local search and evolutionary operator are combined to prevent degradation in the process to generate next population. In LHS-MOEA, LHS local search is used to exploit promising areas and evolutionary operator is used to explore new areas in search space. We achieved two LHS-MOEAs: LHS-MOEA/I and LHS-MOEA/P. Through experiments on 4 CPS_MOPs test problems, the results demonstrate that LHS-MOEA performs much better than traditional MOEA: NSGA-II in solving CPS_MOPs.

Acknowledgements

Authors acknowledge the supports from the National Natural Science Foundation of China under Grant No. 60773047, and the Keystone Science Research Project of the Education Office of Hunan Province of China under Grant No. 06A074.

References

[1] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
[2] Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)
[3] Knowles, J., Corne, D.: Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. IEEE Transactions on Evolutionary Computation 7(2), 100–116 (2003)
[4] Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A Regularity Model Based Multiobjective Estimation of Distribution Algorithm. IEEE Transactions on Evolutionary Computation 12(1), 41–63 (2008)
[5] Deb, K.: Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary Computation Journal 7(3), 205–230 (1999)
[6] Deb, K.: Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany
[7] Deb, K., Sinha, A., Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. In: Proceeding of Genetic and Evolutionary Computation Conference (GECCO 2006), Seattle, Washington, vol. 2, pp. 1141–1148 (2006)
[8] Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 583–592. Springer, Heidelberg (2006)
[9] Mckay, M.D., Beckman, R.J., Conover, W.J.: A Comparison of Three Methods for the Selecting Values of Input Variables in the Analysis of Out Put from a Computer Code. Technometrics 21, 239–245 (1979)
[10] Deb, K., Agrawal, R.B.: Simulated Binary Crossover for Continuous Search Space. Complex Systems 9(6), 115–148 (1995)
[11] Deb, K., Beyer, H.: Self-Adaptive Genetic Algorithms with Simulated Binary Crossover. Evolutionary Computation 9(2), 195–219 (2001)
[12] Deb, K., Goyal, M.: A Combined Genetic Adaptive Search (geneAS) for Engineering Design. Computer Science and Informatics 26(4), 30–45 (1996)
[13] Sierra, M.R., Coello Coello, C.A.: A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization. In: Proceeding Congress on Evolutionary Computation (CEC 2005), Edinburgh, U.K., pp. 65–72 (2005)
[14] Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary computation and convergence to a Pareto front. In: Late Breaking Papers at the Genetic Programming Conf., Madison, WI, pp. 221–228 (1998)
[15] David, H.W., William, G.M.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)

Energy-Predicted Shortest Routing Tree Algorithm in Wireless Sensor Networks

Ming Zhang[1,2], Chenglong Gong[1], Yuan Feng[1], and Chao Liu[2]

[1] Huaihai Institute of Technology, Lian yungang, Jiangsu, 222005, China
lyg690916@163.com, {gcl,fengyuan}@hhit.edu
[2] Nanjing University of posts & Telecommunications Nanjing;Jiangsu, 210003 China
lc96222@126.com

Abstract. Wireless sensor networks (WSNs) consists of unattended sensors with limited storage, energy (battery power) and computational and communication capabilities. Since battery power is the most crucial resource for sensor nodes, the energy prediction and the shortest path is special important. In this paper, we present an energy-predicted shortest routing tree algorithm (EP-SRT) for wireless sensor networks. it improves energy utility by changing the activity of wireless communication module of sensor nodes, energy prediction model and state transition of sensor nodes, while employs clustering and the principle of Prim and Djkstra to build the shortest routing tree to prolong network lifetime. Simulation results show that EP-SRT performs better than MIP and SRT algorithm with high-density deployment and high traffic.

Keywords: Energy-predicted, shortest routing tree, wireless sensor networks.

1 Introduction

Wireless sensor networks are those in which nodes are low-cost sensors that can communicate with each other in a wireless manner, have limited computing capability, and memory and operate with limited battery power. These sensors can produce a measurable response to changes in physical conditions, such as temperature or magnetic field. The main goal of such networks is to perform distributed sensing tasks, particularly for applications like environmental monitoring, smart spaces and medical systems. [1].

There have been extensive studies on routing in wireless networks in recent years. The procotol proposed in [1] is only give a prediction-based energy map for wireless sensor networks, but not consider the shortest routing path. The work proposed in [2] obtains the energy map of sensor networks by using an aggregation-based approach. Energy prediction and routing algorithm was considered in [3], using remainder energy to select routing from source node to Sink, but don't consider the shortest path, easyly have postpone transmittion and waste energy. The Minimum Incremental Power (MIP) in [4] do not consider the optimal routing. Distributed algorithms to form sparse topologies containing minimum-energy routes were proposed in [5], [6]. Shortest path tree algorithm is proposed in [7], but do not consider the energy predicted.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 117–124, 2008.
© Springer-Verlag Berlin Heidelberg 2008

In this paper, we propose the ideas of energy-predicted and the shortest routing tree. There are situations in which a node can predict its energy consumption based on its own past history. If a sensor can predict efficiently the amount of energy it will dissipate in the future, it will not be necessary to transmit frequently its available energy. This node can just send one message with its available energy and the parameters of the model that describes its energy dissipation. With the help of clustering and shortest routing tree to prolong network lifetime, clustering includes partitioning stage and choosing stage, namely, partition the multi-hop network and then chooses cluster-head which is responsible for receiving, sending and maintaining information in its cluster. Then all cluster-heads will build a shortest routing tree to save energy and shorten path. Simulation results show that the system's performance have further improved by using clustering and shortest routing tree than MIP and SRT, It is a promising approach and deserves more future research.

2 Network Model

2.1 Energy Prediction Model

In order to predict the dissipated energy, we studied a probabilistic model based on Markov chains. In this model, each sensor node can be modeled by a Markov chain. In this case, the node operation modes are represented by the states of a Markov chain and, if a sensor node has M operation modes, it is modeled by a Markov chain with M states. Using this model, at each time the node is in state i, there is some fixed probability, Pij, that, in the next time-step, it will be at state j. This probability can be represented by Pij=P { Xm+1=j|Xm=j}. We can also define the n-step transition probability, Pij(n), that a node currently in state i will be in state j after n additional transitions[8] as formula (1):

$$P_{ij}^{(n)} = \sum_{k=1}^{M} P_{ik}^{(r)} P_{kj}^{(n-r)} \quad (<0r<n). \tag{1}$$

With the knowledge of probabilities Pij(n) for all nodes and the initial state of each node, it is possible to estimate some information about the network that can be useful in many tasks. In this work, we will use these probabilities to predict the energy drop of a sensor node. The first step to make this prediction is to calculate for how many time-steps a node will be in state s in the next T time-steps. If the node is in state i, the number of time-steps a node will stay in the state s can be calculated by: $\sum_{t=1}^{T} P_{is}^{(t)}$. Also, if Es is the amount of energy dissipated by a node that remains one time-step in state s, and the node is currently in state i, then the expected amount of energy spent in the next T times, ET(i) as formula (2):

$$E^{T}(i) = \sum_{s=1}^{M} \left(\sum_{t=1}^{T} P_{is}^{(t)} \right) \times E_s \tag{2}$$

Using the value ET(i), each node can calculate its energy dissipation rate△E: △E = △T/T. △E mean the average consumption energy in every time-step for the next T time-steps[7].Each node then sends its available energy and its △E to the monitoring node. The monitoring node maintains an estimation for the dissipated energy at each

node by decreasing the value △E periodically for the amount of remaining energy of each node. The better the estimation the node can do, the fewer the number of messages necessary to obtain the energy information and, thus, the fewer the amount of energy spent[1].

2.2 State Transition Model

The energy dissipation in wireless sensor networks have three model: sensor model; procession model; wireless radio model [9].In order to prolong network lifetime, nodes in the SRTA have various operation modes with different levels of activation and, thus, different levels of energy consumption. We put forward the new state conversion model which have flag of valve which depend on the EPGR state model. In this model, each node has six operation modes: mode 1: sleeping-sensing off and radio off; mode 2: sensing -sensing on and radio off; mode 3: receiving sensing on and radio receiving; mode 4: transmitting -sensing on and radio transmitting; mode 5: listening - sensing on; mode 6: long sleeping- sensing off and radio off for ever ,no responding. The transitions between these modes are described by he diagram of Fig. 1. In that diagram, the operation modes are represented by states sleeping, sensing, receiving, transmitting, determining and long sleeping.

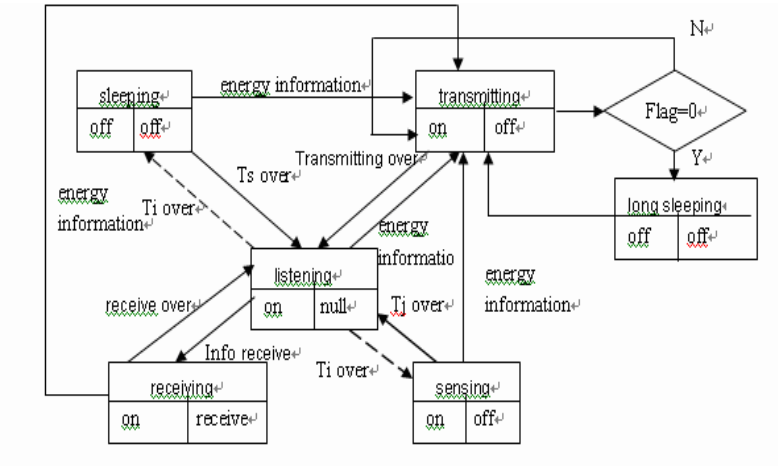


Fig. 1. Node's state transition model. This shows the node's state transition process from one to another. where Ts is a sleeping timer, Ti is listening timer, Tj is a sensing timer.

The diagram of Fig. 1 shows the ''commands'' performed along the path (transition) between states[7]. It means that whenever a node changes its state-based energy dissipation model to current state it performs tests and actions until the new state is reached. "sleeping''—determines whether the node will sleep or not; The ''receiving'' test depends also on the characteristics of the event. Its value is influenced by the degree of cooperation needed by the application. The ''sensing'' test is called only if there is no event in the area of the node. If no event happens, this test will depend on the degree of coverage needed by the application. In the "transmitting" state, if

flag=0, then sensing off and radio off and node convert to long sleeping; if flag=1, then convert to the transmitting. The “listening” test determines whether a new sensing event is present; the long sleeping denotes the node never respond any event. “Timer’’ is an action that starts a timer. The outcome of each test depends on a probabilistic parameter associated with the test. These transitions try to capture the behavior of a sensor node, specially in terms of energy consumption.

3 Energy-Predicted Shortest Routing Tree Algorithm

In this work, we consider the wireless networks where all nodes in the network are homogenous and energy constrained, and each node can directly communicate with any node in given area. In this environment, instead of using a flat configuration, adopting the clustering approach can statistically multiplex many connections into a few paths so that the overall interferences can be reduced with well-controlled access.

3.1 Building Cluster Head

The process of constructing cluster-head is comprised of two phases, as illustrated below:

(1) Initial phase: First, partitions the multi-hop network and then chooses cluster-heads with the following stages[10]:

Partitioning stage: Every node i maintains a triplet: a unique identification ID(i), a cluster identification to which i belongs ,CID(i),and its remaining energy, Crp(i). Such information is exchanged by a MAC layer protocol and thus can be obtained by the neighbors within one-hop distance. By using the first two values, CHEP partitions the multi-hop network into clusters according to some distributed clustering algorithm.

Choosing stage: Every node within a cluster sends its remaining energy Crp(i), to the cluster-head to which it belongs. The cluster-head, upon receiving the power information choose the node with the maximum power as the new cluster-head and broadcasts the decision to its member. From now on, he cluster-head become a normal member, which listens to the messages from the new head just like the other members. (2) Re-clustering phase: When the energy of a cluster-head is lower than threshold, a cluster-head switches its role back to a node with the most residual power within the cluster. Since the power information is maintained by the cluster-head, this can be done easily with broadcasting message, and none of the distributed algorithms, such as those for the initial phase.

3.2 Building Shortest Routing Tree

With the principle of Prim and Djkstra, we set V nodes in WSNs, so we can complete routing tree in |V|-1 step. this network is represented by a undirected graph G= (V,E,W)where nodes in the set V represent the cluster-head and edges (i,j) where $i \in V$ and $j \in V$, in the set E denotes that the two cluster-heads i and j are in the communication range of each other and can transmit a packet to each other. The weight W(i,j) associated with an edge (i,j) $\in$ E indicates the cost(in terms of energy

consumption) of communication between the two cluster-heads i and j. In order to expe-diently explain, we suppose sets of P,S,TE, thereinto, P= {v|v is a node in the shortest routing tree}, S=V-P,TE= {(u,v)|(u,v) is a edge in the shortest routing tree}, suppose $W_{(vi)}$ is the minimum coefficient, namely, is the countdown of the sum from Sink to vi in the shortest tree, $W_{(vi)}$ is save in the cluster-head i[7].

The concrete description of building shortest routing tree is as follows:

(1) Initialization: First solve the shortest tree which Sink is the root, set r=0, $W_{(sink)}=0$, $W_{(vi)}=\infty$, i=1,2,…n.P= {Sink} , S= {v_i|i is 1,2,..n} ,TE= {}.

(2) Find out all edges which W_{ij} is greater than Q, intherinto, $v_i \in P$ and $v_j \in S$,Q is the threshold, then select a edges $W_{i0,j0}$ which Wi0,j0 is maximum(thereinto $v_{i0} \in P$ and $v_{j0} \in S$), and v_{j0} will insert into P. then find out all edges $V_{i,j0}$ which vi is in P and is the neighbor of v_{j0} and connected with v_{j0}, thereinto i=1,2,….k.

(3) Find out the minimum coefficient $W_{(i,j0)}$ which (v_i,v_{j0}) is a edge, then set $P=P \cup \{v_{j0}\}$, $S=S-\{v_{j0}\}$,$TE=TE \cup \{(v_i,v_{j0})\}$.

(4) If S is empty, then end; else r=r+1 and goto (2)

3.3 Energy-Predicted Shortest Routing Tree Algorithm

The Fig2 is the energy-predicted shortest routing tree algorithm:

```
(1)initialization;
(2)build the cluster-head :
   { partition network ;
     choose cluster-head;
   }
(3)if partition is successful then
   { (4) build the shortest routing tree;
     (5) if building is successful then
          { (6)listening;
            (7)if there is a event then
              { (8) processing;
                (9) energy prediction;
                (10) if discretionary node will exhaust then goto (2);
                (11) else goto (6)
              }
          }
     (12) else goto (13)
   }
(13) else end;
```

Fig. 2. Flow chart of energy-predicted shortest routing tree algorithm

4 Simulation Results

In this paper, simulations are given for the algorithm and the results compared with that of MIP and SRT. The following experiments are implemented: the partition of nodes to clusters; energy prediction and building the shortest routing tree. Inthereinto, energy consumption when Eelec is changed from 10nJ/bit to 100nJ/bit and the diameter D of the nodes field is changed from 100m x 100m to 1000m x 1000m; 500 nodes randomly distributed in a square, 500 nodes with equal initial energy and radio

radius are independently generated and randomly distributed, tasks are generated randomly, Their connections are decided by distance. The same topology is used for all simulated protocols (or algorithms) like MIP, STR and EP-SRT. In EP-SRT, according to formula (2) to predict energy, then partition network and building shortest routing tree. The succedent diagrams show that EP-SRT performs better than MIP and SRT algorithm for wireless sensor network with high-density deployment and low traffic.

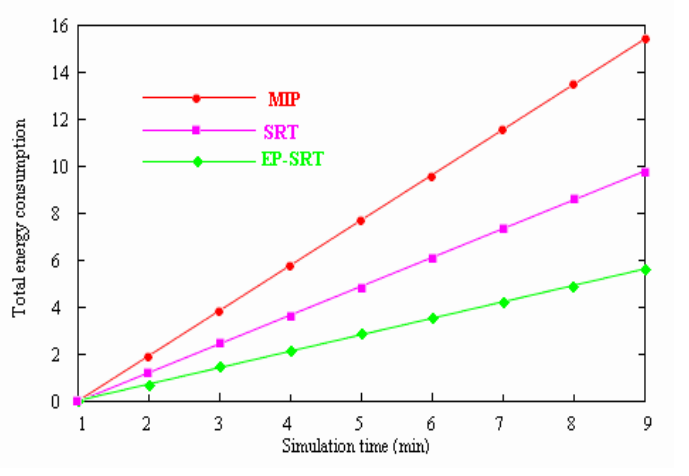


Fig. 3. Total energy consumption of EP-SRT, MIP and SRT

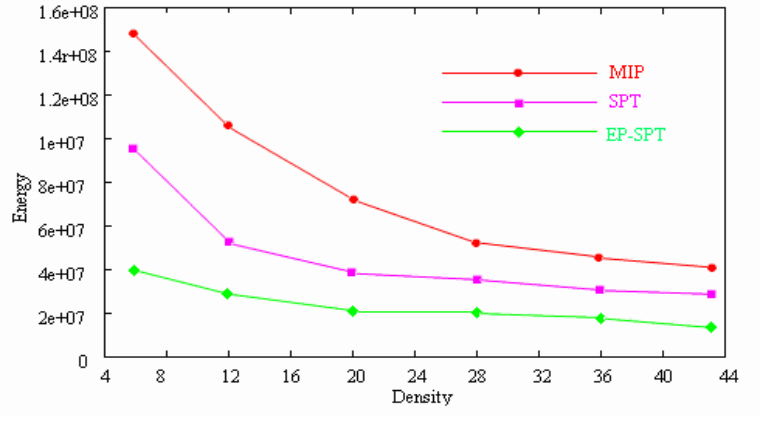


Fig. 4. Total energy consumption of different density.

Fig.3 shows that the total energy consumption is linear growth with the time. we can see that the total energy consumption of EP-SRT is slowly growth, while MIP is quich growth, so EP-SRT algorithm can prolong network lifetime.

From the diagram of Fig. 4, we can see that there are considerable differences in energy consumption among those three algorithms. When the node density is higher, the total energy consumption is lower, EP-SRT can use energy prediction to reduce energy cinsumption to prolong network lifetime.

5 Conclusions

In this paper, we propose the ideas of energy-predicted and the shortest routing tree. There are situations in which a node can predict its energy consumption based on its own past history. If a sensor can predict efficiently the amount of energy it will dissipate in the future, it will not be necessary to transmit frequently its available energy. This node can just send one message with its available energy and the parameters of the model that describes its energy dissipation. With the help of clustering and shortest routing tree to prolong network lifetime, clustering includes partitioning stage and choosing stage, namely, partition the multi-hop network and then chooses clusterhead which is responsible for receiving, sending and maintaining information in its cluster. Then all cluster-heads will build a shortest routing tree to save energy and shorten path. Simulation results show that the system's performance have further improved by using clustering and shortest routing tree than MIP and SRT, It is a promising approach and deserves more future research.

Acknowledgements

We acknowledge the support of High-Tech Research plan of Jiangsu Province (No.BG2007045), we sincerely thank the anonymous reviewers for their constructive comments and suggestions.

References

1. Mini, R.A.F., do Val Machado, M., Loureiro, A.A.F., Nath, B.: Prediction-based Energy Map for Wireless Sensor Networks. Ad Hoc Networks 3, 235–253 (2005)
2. Zhao, Y.J., Govindan, R., Estrin, D.: Residual energy scans for monitoring wireless sensor networks. In: Proceedings of WCNC, Orlando (2002)
3. Lin, K., Zhao, H., et al.: Energy Prediction and Routing Algorithm in Wireless Sensor Networks. Journal on Communications 27(5), 21–23 (2006)
4. Banerjee, S., Misra, A., Yeo, J., Agrawala, A.: Energy-Efficient Broadcast and Multicast Trees for Reliable Wireless Communication. In: Proc. IEEE Wireless Communications and Networking Conference, pp. 660–667 (2003)
5. Rodoaplu, V., Meng, T.H.: Minimum Energy Mobile Wireless Networks. IEEE J. Select. Areas Communi. 17(8), 1333–1334 (1999)
6. Li, L., Halpen, J.Y.: Minimum Energy Mobile Wireless Networks Revisited. In: IEEE International Conference on Communications (ICC) (2001)
7. Yao, Z.-h., Wang, G.-c., Sun, Y., Qiu, J.-l.: A Routing Algorithm with Low-Delay Energy-Efficient base on SPT-in Sensor Network. Microelectronics and Computer 24(9), 174–180 (2007)
8. Ross, S.: A First Course in Probability, 5th edn. Prentice-Hall, Englewood Cliffs (1998)

9. Heinzelman, W.R., Balakrishnan, A.C.H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proceeding of HICSS 2000, Hawaii, USA, vol. 2, pp. 4–7 (January 2000)
10. Liu, J.-S., Lin, C.-H.R.: Energy-efficiency clustering protocol in wireless sensor networks. Ad Hoc Networks (3), 371–388 (2005)

Evolutionary Antenna Design via Modified Normalized GT Algorithm

Yuanyuan Fan[1,2], Qingzhong Liang[2], Zhenhua Cai[2], and Hui Li[2]

[1] School of Computer Science, Wuhan University, China
430072 Wuhan, China
yyfan@cug.edu.cn
[2] School of Computer Science, China University of Geosciences, China
430074 Wuhan, China

Abstract. Jason Lohn and his workmates in NASA Ames Research Center have evolved an X-band antenna for NASA's Space Technology 5 (ST5) mission which is the first evolved hardware in space. However their approach did not put too much attention on the efficiency of the evolutionary algorithm. Owing to the flaw we employ GT Algorithm to tackle constraints and balance multi-objective via normalization which makes evolution more efficient. Moreover we adopt a kind of linear real-values code to describe the structure of antenna so that it is easier to carry out genetic operations and control the size of the antenna. We have evolved a wire antenna successfully via this approach and all the targets have been meted.

Keywords: evolutionary antenna design, constrained multi-objective optimization, evolutionary algorithm, GT algorithm.

1 Introduction

Communication, radar and remote sensing systems employ thousands of different types of antennas, and it is increasingly important that antennas should be high-performance, customized and with lower design cost. But traditional methods of antenna design by hand are time- and labor-consuming with requirement of vast knowledge and experience on antenna design but limit complexity. Besides, even with the aid of simulation technique and local optimization methods, the designers must provide an initial prototype that should be very close to the final design. While evolutionary algorithms start form several random designs without any initial guess and require the minimal amount of information of design parameters. In addition evolutionary algorithms are robust enough to search large design space without tapped into local optimum which would easily occur in many traditional methods. Therefore evolutionary algorithms are quite promising for automated antenna design.

Automated antenna synthesis via evolutionary algorithm has garnered more and more attention and investigation since early 1990s, [1], [2], [3], including wire antennas[4], antenna arrays[5], helical antennas[6], reconfigurable antenna[7] and so on. Some of them are so novel and irregular that they seem unlikely to be achieved by human beings.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 125–133, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Specially, on March 22, 2006, three Ames-developed evolved antennas were launched as part of NASA's Space Technology 5 (ST-5) mission, and on June 30, 2006 the Goddard Space Flight Center (GSFC) -based mission completed operations with the "flawlessly" performance of the antennas. It represents the first time an artificially evolved object has flown in space which is one of the most convincing demonstrations of using artificial intelligence to automatically design a useful structure. In this project a kind of branching GA [8] was used. However there was not too much attention centered on the efficiency of algorithm. Their tree coding method is a little too complex so as to tend toward invalid individuals and make genetic operators inefficient. Besides there are too many parameters required to be tuned. So we would focus more attention on increasing the efficiency of our algorithm.

As many other real world optimization problem, antenna design is a kind of constrained multi-objectives optimization problem involving several inequality and/or equality constraints. The penalty methods are very popular to handle the constraints, but it is difficult to find appropriate penalty parameters or constraints priority to guide the search towards global optimum. [9], [10], [11] While GT Algorithm [12] adapt to such kind of constrained optimization. Therefore we adopt GT algorithm to tackle the constraints by adding discrete crossover and mutation to enhance its exploring ability. Besides the management of multi-objectives is another difficulty in solving the antenna optimization problem. In our approach we turn multi-objectives into a single objective by normalizing all the values of objective function. Upon that we name the algorithm Modified Normalized GT Algorithm (MNGTA).

Remaining parts of this paper will be organized as below: part 2 describes the details of MNGTA, part 3 is our approach for the ST5 antenna design via MNGTA, part 4 includes the results of our experiment, and part 5 presents the conclusion.

2 Modified Normalized GT Algorithm (MNGTA)

A constrained multi-objective problem can be described or easily transformed into the form as below:

$$
\begin{aligned}
&\textit{Maximize} \quad f_1(\vec{x}), f_2(\vec{x}),..., f_m(\vec{x}) \\
&\textit{Subject to} \\
&\quad g_j(\vec{x}) \le 0 \qquad j = 1,2,...,q \\
&\quad h_j(\vec{x}) = 0 \quad j = q+1, q+2,...,k \\
&\quad X = \{\vec{x} = (x_1, x_2, \ldots, x_n) \mid l_i \le x_i \le u_i\} \\
&\quad \vec{l} = (l_1, l_2, \ldots, l_n) \\
&\quad \vec{u} = (u_1, u_2, \ldots, u_n)
\end{aligned}
\tag{1}
$$

Here $\vec{x}$ is the decision variable, while $\vec{l}$ and $\vec{u}$ are the lower and upper limits of $\vec{x}$ respectively.

2.1 The Main Framework

MNGTA adopts the framework of common steady state genetic algorithm as follows:

Randomly create a population P(t) of size N, and set the generation counter t=0;

REPEAT

For each individual $\vec{x}_i$ in P(t) , i=1, 2,...,n:

{

execute offspring generation operator with output $\vec{s}_i$;

compare $\vec{s}_i$ with $\vec{x}_i$. If $\vec{s}_i$ is not worse than $\vec{x}_i$, replace $\vec{x}_i$ with $\vec{s}_i$;

}

t=t+1;

UNTIL evolution ends;

Print population and t.

2.2 Genetic Operator

Here the multi-parent crossover of GT Algorithm is adopted, at the same time modified by adding discrete crossover and mutation. The genetic operator to generate an offspring $\vec{s}_i$ of $\vec{x}_i$ is as follows:

Step1. Randomly choose M different individuals $\vec{x}_{p1}, \vec{x}_{p2}, \ldots, \vec{x}_{pM}$ from population P(t), which are all different from $\vec{x}_i$

Step2. Multi-parent Crossover:

Randomly create with $a_1 + a_2 + \ldots + a_M = 1$, in range $-0.5 \le a_1, a_2, \ldots, a_M \le 1.5$,

Let $\vec{s}_i = a_1 \vec{x}_{p1} + a_2 \vec{x}_{p2} + \ldots + a_M \vec{x}_{pM}$

Step3. Denote $\vec{s}_i = (s_{i1}, s_{i2}, \ldots, s_{in})$, $\vec{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, p_c crossover probability, and p_m mutation probability. For each gene position j ($1 \le j \le n$), do discrete crossover and mutation:

Discrete Crossover: $s_{ij} = x_{ij}$ with p_c;

Mutation: $s_{ij} = s$ with p_m, where s is a random real number $\in [l_j, u_j]$

Step4. The offspring created is $\vec{s}_i = (s_{i1}, s_{i2}, \ldots, s_{in})$.

2.3 Management of Constraints and Multi-Objectives

In order to tackle several difficult constraints, we calculate the total violation. The total violation of individual $\vec{x}_i$ is calculated as follows:

For each constraint j,

$$Violation_j(\vec{x}_i) = \begin{cases} g_j^*(\vec{x}_i), & j = 1,2,\ldots,q \\ \left|h_j(\vec{x}_i)\right|, & j = q+1, q+2, \ldots, k \end{cases} \tag{2}$$

$$g_j^*(\vec{x}_i) = \max\{0, g_j(\vec{x}_i)\} \tag{3}$$

$$Violation(\vec{x}_i) = \sum_{j=1}^{k} Violation_j(\vec{x}_i) \tag{4}$$

The less violation is, the more excellent the individual $\vec{x}_i$ is.

On the other hand, to balance the different objectives we would design a single fitness function by normalizing all the objectives instead of using their function values directly.

$$Fitness(\vec{x}_i) = \prod_{r=1}^{m}(1 + \frac{f_r(\vec{x}_i) - Min_f_r}{Max_f_r(\vec{x}_i) - Min_f_r + \delta}) \tag{5}$$

$$Min_f_r = \min_{\vec{x} \in X} f_r(\vec{x}) \tag{6}$$

$$Max_f_r = \max_{\vec{x} \in X} f_r(\vec{x}) \tag{7}$$

Here m is the number of objectives, and δ is a very small positive value to avoid being divided by zero when Min_f_r equals Max_f_r. After breeding offspring, the minimal and maximal values of all the objectives should be renewed which should occur in each generation.

2.4 Comparison of Individuals

The comparison is based on the total violation and fitness. Considering two individuals $\vec{x}_i$ and $\vec{x}_j$, we define $\vec{x}_i$ is better than $\vec{x}_j$ if any of the following conditions is satisfied.

a. $Violation(\vec{x}_i) < Violation(\vec{x}_j)$ or

b. $Violation(\vec{x}_i) = Violation(\vec{x}_j)$ and $Fitness(\vec{x}_i) < Fitness(\vec{x}_j)$

3 ST5 Antenna Design via NMGTA

The researchers of NASA Ames Research Center have designed an evolved X-band antenna for NASA's Space Technology 5 (ST5) mission in 2003 which has became the first evolved hardware in space. According to the requirement of ST5, we present a more laconic approach and get satisfactory results.

3.1 Requirements of ST5

Table 1 presents the key requirements of ST5 antenna. Voltage standing wave radio (VSWR) is the radio between the highest and lowest voltage in the signal envelope along a transmission line. It is a way to quantify the amount of impedance mismatch at the junction.

Table 1. Key ST5 Antenna Requirements [13]

Property	Specification
Transmit Frequency	8470 MHz
Receive Frequency	7209.125 MHz
VSWR	<1.2: 1 at Transmit Freq <1.5: 1 at Receive Freq
Gain Pattern	≥0dBic, 40< θ ≤80, 0≤ Φ ≤360 ≥-5dBic, 0≤ θ ≤40, 0≤ Φ ≤360
Input Impedance	50 Ω
Diameter	<15.24cm
Height	<15.24cm
Antenna Mass	<165g

3.2 Code of Antenna

Antenna structure starts with an initial feeder and several conductor wires following. We use a linear real-values code to represent antenna geometric structure, since each wire can be described as coordinates of its two terminals. For each wire its start point is the same as the end point of the former wire. Besides we need a variable to represent the radius of the wires. For convenience, all the radii are the same. A certain chromosome is $\vec{x}$=(r, z_0, x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4, x_5, y_5, z_5, x_6, y_6, z_6). The corresponding geometric structure of antenna can be described as Table 2.

Table 2. Mapping of chromosome to geometric structure of antenna

	Start Point	End Point	Radius
Feeder	(0, 0, 0)	(0, 0, z_0)	r
Wire 1	(0, 0, z_0)	(x_1, y_1, z_1)	r
Wire 2	(x_1, y_1, z_1)	(x_2, y_2, z_2)	r
Wire 3	(x_2, y_2, z_2)	(x_3, y_3, z_3)	r
Wire 4	(x_3, y_3, z_3)	(x_4, y_4, z_4)	r
Wire 5	(x_4, y_4, z_4)	(x_5, y_5, z_5)	r
Wire 6	(x_5, y_5, z_5)	(x_6, y_6, z_6)	r

According to the requirements about the geometric structure of antenna, the decision space can be defined as X={ $\vec{x}$ | 0.0003≤r≤0.0005, 0.001≤z_0≤0.002, −0.07≤x_i, y_i≤0.07, 0.001≤z_i≤0.1}, whose unit is metric. With this code we can easily control the size of antenna and carry out genetic operators.

3.3 Constraints and Objectives

According to the requirements of ST5 antenna, the two constraints are

$$\begin{cases} T_VSWR < 1.2 \\ R_VSWR < 1.5 \end{cases} \tag{8}$$

T_VSWR and R_VSWR respectively indicate the VSWR at transmit and receive frequency.

We take the performance of gain as both constraints and objectives.

The constraints on the gain are:

$$\begin{cases} \min_T_Gain_0_40 \geq -5 \\ \min_T_Gain_40_80 \geq 0 \\ \min_R_Gain_0_40 \geq -5 \\ \min_R_Gain_40_80 \geq 0 \end{cases} \tag{9}$$

$$\min_T_Gain_0_40 = \min_{0^\circ \leq \theta \leq 40^\circ, 0^\circ \leq \varphi \leq 360^\circ} T_Gain_{\theta\varphi} \tag{10}$$

$$\min_T_Gain_40_80 = \min_{40^\circ < \theta \leq 80^\circ, 0^\circ \leq \varphi \leq 360^\circ} T_Gain_{\theta\varphi} \tag{11}$$

$$\min_R_Gain_0_40 = \min_{0^\circ \leq \theta \leq 40^\circ, 0^\circ \leq \varphi \leq 360^\circ} R_Gain_{\theta\varphi} \tag{12}$$

$$\min_R_Gain_40_80 = \min_{40^\circ < \theta \leq 80^\circ, 0^\circ \leq \varphi \leq 360^\circ} R_Gain_{\theta\varphi} \tag{13}$$

The formula (10) ~ (13) indicate the minimum power gain in specific direction indicated by φ, θ angles in sphere coordinate at transmit and receive frequencies.

The two objectives on the gain are Maximization of both T_Gain and R_Gain, which respectively indicate the sum of power gain in certain directions ($40^\circ \leq \theta \leq 80^\circ$) at transmit and receive frequencies.

$$T_Gain = \sum_{\varphi=0^\circ}^{360^\circ} \sum_{\theta=40^\circ}^{80^\circ} T_Gain_{\theta\varphi} \tag{14}$$

$$R_Gain = \sum_{\varphi=0^\circ}^{360^\circ} \sum_{\theta=40^\circ}^{80^\circ} R_Gain_{\theta\varphi} \tag{15}$$

Then the ST5 antenna design can be transformed into the problem of maximize T_Gain and R_Gain subject to formula (8) and (9).

3.4 Parameters Setup and Simulator

Population size: N=100
Number of Parents: M=5
Crossover probability: p_c =0.1
Mutation probability: p_m =0.05

Stop condition: the violation of the best individual equls to zero and the number of evaluations is less than the maximum number of evaluations which is 50,000.

We use Numerical Electromagnetic Codes, Version 2 (NEC2) [14], [15] to simulate all antenna designs. An infinite ground plane approximation was used in the simulation. It was found to provide enough accuracy while accelerating evaluation.

4 Results of the Experiment

We run MNGTA several times and get quite a few results. Here we only presented one of them on account of the space limitation. The antenna structure is presented in figure 1. The VSWP is shown in figure 2. In figure 3, 4 we give the curves of the minimum and maximum gains at transmit and receive frequencies respectively. The results has revealed that the evolved antenna meet all of the requirements of ST5 mission. The evolved antenna achieves high gain across a wide range of elevation angles which would result in less power being required. In addition both VSWR of the antenna at transmit and receive frequencies are such small that it do not require a matching network or phasing circuit.

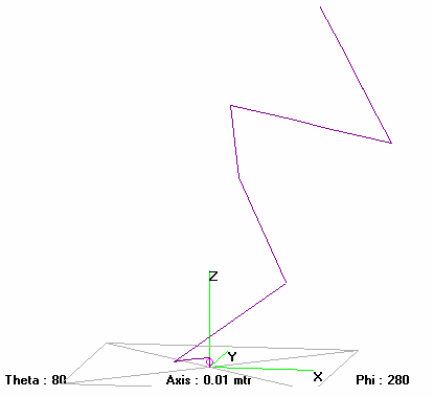


Fig. 1. The structure of one of the evolved antennas

However we have not build and test a prototype of the evolved antenna in real environment. But with the enough accuracy of the NEC the experiment has strongly proved that MNGTA is capable of resolving the real antenna design problems in our future work.

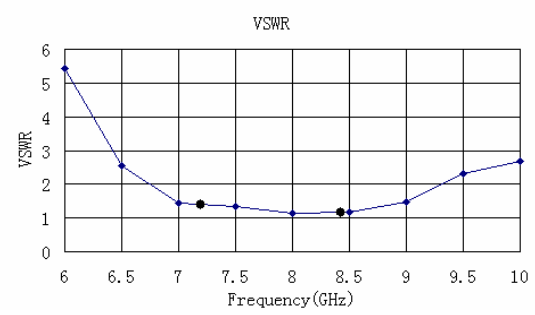


Fig. 2. The VSWR of the evolved antenna in Figure 1

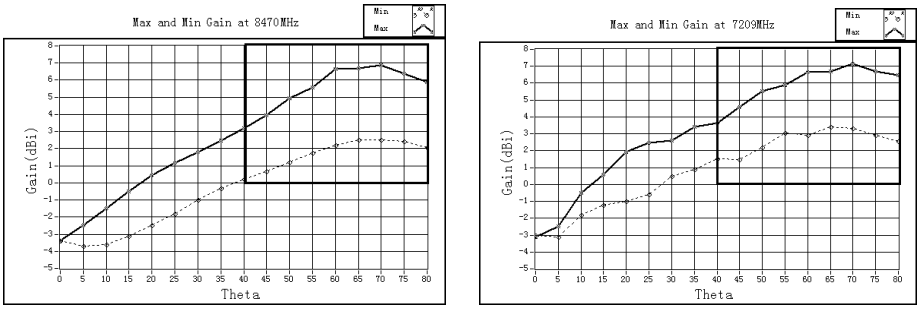


Fig. 3. The maximum and minimum gain at 8470MHz and 7209MHz of the evolved antenna

5 Conclusion

We propose a novel evolutionary algorithm MNGTA to tackle the problem of ST5 antenna design which is a constrained multi-objective problem. The result of our experiment shows that the evolved antennas via MNGTA meet all the requirements of ST5 mission.

In contrast to the approach of NASA, our algorithm has some advantages as follows:

Firstly, the linear real-values code we adopt is more laconic so that it is easier to carry out the genetic operations such as initialization, crossover, mutation, and so on without any tendency towards invalid individual. At the same time, this coding method is convenient to control the size of the antenna.

Secondly, the management on the constraints and stop condition make the result we get would meet all of the requirements of ST5 for sure.

Lastly, there is no parameter or priority to be tuned via MNGTA which makes it easily employed to other constrained multi-objective optimization problem almost without any modification. Moreover it would not consume much time to tune the parameters.

Acknowledgments. This work was supported by The Natural Science Foundation of China University of Geosciences (No:CUGQNL0843), and by the open research program of the Geological Processes and Mineral Resources (GPMR) ,China University of Geosciences(No. GPMR200618).

References

1. Michielssen, E., Sajer, J.-M., Ranjithan, S., Mittra, R.: Design of lightweight, broad-band microwave absorbers using genetic algorithms. IEEE Transactions on Microwave Theory and Techniques 41(6), 1024–1031 (1993)
2. Haupt, R.L.: An introduction to genetic algorithms for electromagnetics. Antennas and Propagation Magazine, IEEE 37(2), 7–15 (1995)
3. Linden, D.S., Altshuler, E.E.: Automating Wire Antenna Design using Genetic Algorithms. Microwave Journal 39(3) (March 1996)
4. Linden, D.S., Altshuler, E.E.: Evolving Wire Antennas Using Genetic Algorithms: A Review. In: Proceedings of the First NASA/DoD Workshop on Evolvable Hardware, pp. 225–232 (1999)
5. Globus, A., Linden, D., Lohn, J.: Evolutionary Design of a Phased Array Antenna Element. In: Antennas and Propagation Society International Symposium 2006, July 9-14, pp. 2071–2074. IEEE, Los Alamitos (2006)
6. Lohn, J.D., Kraus, W.F., Linden, D.S.: Evolutionary optimization of a quadrifilar helical antenna. In: Antennas and Propagation Society International Symposium, vol. 3, pp. 814–817. IEEE, Los Alamitos (2002)
7. Linden, D.S.: In-situ evolution of a reconfigurable antenna. In: IEEE Proceedings of Aerospace Conference, vol. 5, pp. 2333–2338 (2001)
8. Space Technology 5 Mission, `http://nmp.jpl.nasa.gov/st5/`
9. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (2000)
10. Takahama, T., Sakai, S.: Constrained Optimization by the ε Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites. In: IEEE Congress on Evolutionary Computation, CEC 2006, July 16-21, pp. 1–8 (2006)
11. Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello, C.A.C., Deb, K.: Problem definitions and evaluation criteria for the CEC2006 special session on constrained real-parameter optimization (2006)
12. Guo, T., Kang, L.: A new evolutionary algorithm for function optimization. Wuhan University Journal of Natural Sciences 4(4) (1999)
13. Lohn, J.D., Linden, D.S., Hornby, G.S., Rodriguez-Arroyo, A., Seufert, S.E., Blevins, B., Greenling, T.: Evolutionary Design of a Single-Wire Circularly-polarized X-Band Antenna. In: Antennas and Propagation Society International Symposium, vol. 2B, pp. 267–270. IEEE, Los Alamitos (2005)
14. Numerical Electromagnetics Code NEC2 unofficial home page, `http://www.nec2.org/`
15. NEC-2 Manual, Part III: User's Guide, `http://www.nec2.org/other/nec2prt3.pdf`

Quantitative Cryptanalysis of Six-Round DES Using Evolutionary Algorithms

Fan Yang[1,2], Jun Song[1,2], and Huanguo Zhang[2]

[1] Computer School, China University of Geosciences, Wuhan 430074, China
[2] Computer School, Wuhan University, Wuhan 430079, China
planesail@163.com

Abstract. This paper presents a new approach integrated evolutionary optimization into cryptanalysis techniques. The idea that relies on key likelihood estimation and linearity approximation is used to design the fitness function model. The experimental results indicate that the above techniques are effective to attack the 49 bits key and the 42 bits key for DES cipher reduced to six rounds or less. Furthermore, preliminary analyses indicate that the designing of fitness function and the capability of resisting linear approximation are the key factors affecting evolutionary cryptanalysis.

Keywords: Evolutionary computation, Evolutionary cryptanalysis, Likelihood estimation, Linear approximation.

1 Introduction

The cryptanalysis technique adopting evolutionary computation (Evolutionary Cryptanalysis) is the combination of evolutionary computation and cryptography, which is the developing tendency for the automatization of cryptanalysis. In recent years, many researchers investigate in the field of evolutionary cryptanalysis. In 1993, Spillman etc. and Forsyth etc. showed that monoalphabetic ciphers could be attacked using Genetic Algorithm (GA) [1] and Simulated Annealing Algorithm (SA) [2] respectively. Later, many researchers such as Clark etc. [3] in 1998, Grundlingh etc. [4] in 2002, Uddin etc. [5] in 2006 introduced Tabu Searching Algorithm (TS) and PSO (Particle Swarm Optimization) to that cryptanalysis. Clark etc. [6] in 1998 and Dimovski etc. [7] in 2003 applied Parallel Genetic Algorithm (PGA) in the cryptanalysis of polyalphabetic cipher Vigenère. For the cryptanalysis of the simple transposition ciphers, Clark etc. [8] in 1994 and Dimovski etc. [9] in 2003 adopted GA/TS/SA, while Russell etc. and Toemeh used Ant Colony Algorithm (AC) [10] in 2003 and GA [11] in 2007 separately. In 2006, Nalini adopted the intelligent analysis on S-DES cipher by the GA/TS/SA methods [12].

Clark analyzed and described the research status and developing trend of intelligent cryptanalysis systematically in 2004 [13]. From the existent achievements, intelligent technique is mainly applied to the cryptanalysis of classical ciphers or simple cipher algorithms [13], the primary reason for which is that the designs of fitness function for classical ciphers are based on the simple linearity functions or language statistic

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 134–141, 2008.
© Springer-Verlag Berlin Heidelberg 2008

property, easy for linearity approximation and mathematical model-building. However, designs for modern ciphers generally meet the avalanche and the diffusion rule and the key scheme is independent of the encryption and decryption process, so it is hard to construct the mathematical models through plaintext, ciphertext and the cipher features. Consequently, intelligent cryptanalysis for modern cryptography is quite a difficult and long-term task [13].

This paper presents a model of fitness function relying on key likelihood estimation and linearity approximation, by which method we quantitatively analyze and experiment on part keys whose length is 49 and 42 bits respectively in six rounds or less of DES. We preliminarily demonstrate and analyze the impact of DES encryption with different rounds on evolutionary cryptanalysis. Besides, we also present the complexity of implementing brute force attack. The experimental results show that the design of fitness function and the capability of resisting linear approximation are the crucial factors for the effects of the evolutionary cryptanalysis. Compared with the reference [14], our results are obviously superior.

2 Feistel Network and DES Cipher

2.1 Feistel Network and DES Cipher

Feistel network transforms any function (also called Round Function) into a permutation [15] and this structure is adopted by many block ciphers, such as DES, Camellia, RC6 and Twofish etc. The typical features are presented below: the structure is composed of many round functions and the decryption algorithm is the same as the encryption one except that the key scheme is used in the reverse order.

The encryption process of r-round Feistel cipher with $2n$-bit block length is as follow:

(1) Given the plaintext P, as $P=L_0R_0$. L_0 and R_0 are the left and right n bits of P respectively.

(2) Repeat the same operations of r rounds while combining data with key in every round and L_iR_i ($1\leq i\leq r$) can be gained according to the following rules:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_i \oplus F(R_{i-1}, K_i) \end{aligned} \tag{1}$$

Where $\oplus$ means the xor operation of two bit strings, F: $GF(2)^n \times GF(2)^n \rightarrow GF(2)^n$ is the round function and K_i is the subkey generated by the seed key K.

(3) Output the ciphertext $C=L_rR_r$. In the last round, permutation of the left and right shouldn't be taken.

DES is the most widely used block cipher so far. By encrypting 64-bit plaintext with 56-bit key, 64-bit ciphertext is created. 56-bit seed key K generates sixteen 48-bit subkeys K_i used in sixteen rounds. The decryption is performed adopting the same algorithm while using the subkeys K_i in the reverse order.

2.2 Pattern Theorem and the Convergence of Evolutionary Algorithm

Definition 1 [16]**.** Pattern Theorem. The pattern H is called Pattern Order of the pattern, recorded as $O(H)$. The distance of the first and last positions bearing certain gene values in pattern H is defined as Pattern Definition Length, recorded as $\delta(H)$.

Theorem 1. Under the collaboration of selection, mating and mutation operator, the genetic algorithm of Pattern Theorem [16] obtains the low-order, short definition length. Moreover, the average fitness values will increase at exponential rate on the condition that the values are higher than the swarm average ones.

Theorem 2 [16]**.** The probability that the basic genetic algorithm converges in the optimal solution is less than 1.0.

Theorem 3 [16]**.** The probability that the genetic algorithm using the preserved optimal individual converges in the optimal solution is 1.0.

Theorem 4**.** If there is a transformation (or intelligent optimization model) $G\{gk': M \rightarrow C\}$ being the fitness function f, the test key k' can be optimized. For N plaintext-ciphertext pairs, given the set of all optimized keys $K'=\{k_1', k_2', k_3', \ldots, k_N'\}$, if the optimized key k_i' can be selected from K' at the probability ρ so that $\Psi(ki') > t$ for any k_i' in which t is a varying integer according to research objects of different ciphers ($1 \leq t \leq n$), the complexity of implementing brute force attack on k_i' is less than $\rho \cdot N \cdot C_n^{n-t} \cdot 2^{n-t}$.

2.3 Evolutionary Cryptanalysis on 14-Bit Key of 4-Round DES

The research productions and correlative documentations about DES evolutionary cryptanalysis are relatively few. Laskari etc. [14] put up the intelligent cryptanalysis on 14-bit keys of 4-round DES using PSO technique in 2005, while the corresponding search space of key is 2^{14}.

Biham and Shamir proved in 1991 that 42 bits out of 56 bits can be gained by differential cryptanalysis, while the left 14 bits can be found by brute force attack on the condition that the probability about 1-round differential feature of 4-round DES is 1.0[17]. Laskari etc. took evolutionary cryptanalysis on the above-mentioned 14-bit keys by PSO method [14], transforming brute force attack of the left 14-bit keys into intelligent search.

The design of Laskari's fitness function is as follow [14]: Given 42 bits out of key's 56 bits, suppose X is a 14-dimension vector, i.e. a test key of 14 bits. Suppose that np is the amount of ciphertext pairs, decrypt np ciphertext pairs and calculate cnp_X: the amount of the decryption pairs meeting the plaintext-known differential condition. The fitness function f satisfies:

$$f(X) = np - cnp_X \tag{2}$$

Based on the above fitness model, the average success rate of the experiments is approximately 99.3% when adopting 6 distinct keys and np is 20(50).

3 GA Cryptanalysis on DES Cipher

3.1 Design of the Fitness Function

Suppose that M is the set of plaintexts, C is the set of ciphertexts, K is the set of keys, the real key is k, the test key is k' where $k, k' \in K$. Given any plaintext block $M_i \in M$, the relevant ciphertext under the control of the real key k is C_i^k, while the ciphertext generated by the test key k' is $C_i^{k'}$ in which $C_i^k, C_i^{k'} \in C$. L is the length of keys and the fitness function f is defined as:

$$f(M_i, k') = (L - \sum_{j=1}^{B} H(C_i^k, C_i^{k'})) / L \tag{3}$$

H denotes the Hamming Distance of the relevant binary strings, which depicting the different bit amount of C_i^k and $C_i^{k'}$. $B=L/8$, i.e. the byte amount of L. For DES, L means 56 bytes.

The design of above fitness function bears certain universality, applicable for the evolutionary cryptanalysis on other block ciphers. The Fitness Landscape (**3**) is accidented leading to many partial optimal solutions. The value of fitness is not the criteria for distinguishing the good or bad of the individuals in the swarm strictly. The value can achieve the overall maximum 1.0 when k and k' are equivalents.

3.2 Definition of the Evolutionary Parameters

The evolutionary cryptanalysis on DES encodes in binary and every key embodies an individual. The swarm is initialized in random way. As for the genetic operator, the selection operation *Slt* adopts the optimal strategy, the crossover operation *Crs* adopts multi-point crossover way and the mutation operation *Mut* adopts multi-point mutation mode in which the crossover rate is between the area of 0.70 and 0.80 and the mutation rate is 0.001. The swarm size is N, the generations of running is R and the amount of the plaintext-ciphertext pairs is G.

3.3 GA Cryptanalysis Algorithm of DES

Input: plaintext-ciphertext pairs: M_i and C_i, where $M_i \in M$ and $C_i \in C$, $1 \le i \le G$
Output: G result keys: $k'_{prox_1}, k'_{prox_2}, \ldots, k'_{prox_G}$

```
for(1 ≤ i ≤ G)
begin
Input(M_i, C_i);
  for(1 ≤ j ≤ R)
  begin
   Initialize(X);
   fitness(M_i, X_i);
```

```
    do
            Select(X);
            Crossover( );
            Mutation( );
            fitness(M_i, X_i');
       while((j = j + 2)&&(1 • j • N))
     end
     Output(k'_prox_i)
end
```

In the above algorithm, X_i means the individuals in the swarm X, $X_i' \in X$, $X_i' = \langle x_{i1}', x_{i2}', \cdots, x_{im}' \rangle$, $1 \leqslant m \leqslant N$. The result keys $k'_{prox_1}, \cdots, k'_{prox_G}$ are created by G different plaintext-ciphertext pairs (M_i, X_i) in which the individuals the fitness value equals 1.0 are real keys or their equivalent keys.

4 Experimental Results and Analysis

According to Pattern Theorem, we fix the last r bits of chromosome to low down the impact of the non-linear components on the evolutionary cryptanalysis and improve the reliability of the analysis results in DES cipher. Meanwhile, in order to compare and analyze the effects of the evolutionary cryptanalysis with different search complexity, we experiment quantitatively on the left bits with 49 bits (r=8) and 42 bits (r=16) respectively. The results are listed below.

The distributions of the evolutionary cryptanalysis results in the first to the sixth rounds of DES are shown in Figure 1 (r=8) and Figure 2 (r=16) where N =200 and R=1000.

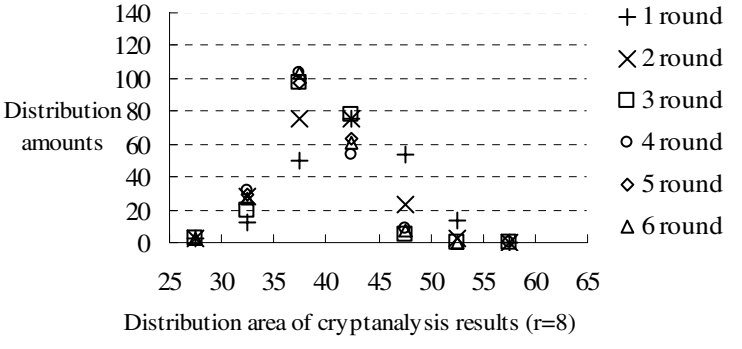


Fig. 1. The distribution of results(r=8)

Figure 1 and Figure 2 depict the distributions of the optimum keys in the first to the sixth rounds where r=8 and r=16. From these figures, we will find that with the increase of the round ordinal, the individual amount of the optimum keys approximating the real

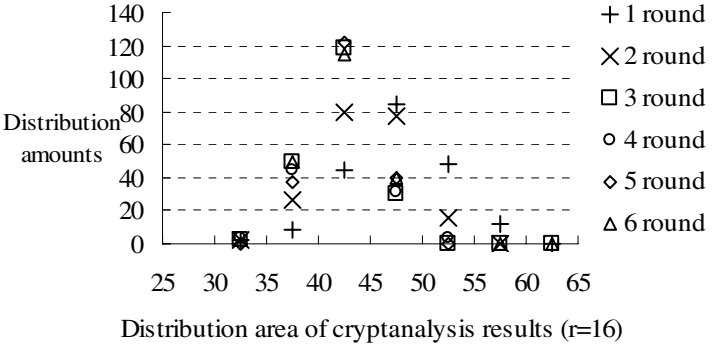


Fig. 2. The distribution of results(r=16)

keys takes on an overall downtrend and the distribution of the optimum keys inclines to normal distribution gradually.

The selection probabilities of the optimum keys in the evolutionary cryptanalysis results of the first to the sixth rounds are listed in Table 1 (r=8) and Table 2 (r=16) where N =200 and R=1000. As showed in Theorem 4, t_1 is a chosen integer for the fitness function f and the probability ρ_1 meets the conditional probability of the optimum keys for any k_i' ($k_i' \in K'$, $\Psi(k_i') > t_1$). The parameters t_2 and ρ_2 are similar.

Table 1. The probability distribution of optimum keys(r=8)

Round	Fitness f	t_1	Probability ρ_1	t_2	Probability ρ_2
1	1.0000	50	0.3125	45	0.75
2	0.9375	50	0.25	45	0.375
3	0.8594	45	0.125	40	0.625
4	0.8438	45	0.1	40	0.433
5	0.8281	45	0.0937	40	0.437
6	0.8281	45	0.094	40	0.481

Table 2. The probability distribution of optimum keys(r=16)

Round	Fitness f	t_1	Probability ρ_1	t_2	Probability ρ_2
1	1.0000	55	0.234	50	0.5107
2	0.9375	50	0.5	45	0.6875
3	0.8594	45	0.3079	40	0.692
4	0.8438	45	0.2903	40	0.903
5	0.8281	45	0.3157	40	1.000
6	0.8281	45	0.2778	40	0.916

Table 1 and Table 2 describe the selection probability of the optimum keys in the first to the sixth rounds where r equals 8 and 16 respectively. These tables, showed that the probability of obtaining the optimum keys approximating the real keys, with the

increase of the round ordinal, tends to decline on the whole and the fitness value of the optimum keys also presents a downtrend.

From the analysis above, some valueable result will be described as follows. First, as the round ordinal of DES algorithm increases, the ciphertext's capability of resisting linearity approximation strengthens and the difficulty of the evolutionary cryptanalysis augments. Second, the design of fitness function in evolutionary cryptanalysis should reflect not only the relationship between different plaintext-ciphertext pairs and the corresponding keys, but also the trend of key overall optimization. Subsequently, the design of fitness function and the capability of resisting linearity approximation for block ciphers are the significant factors affecting the effects of evolutionary cryptanalysis.

5 Conclusion

As the non-linear technique is introduced into the cryptography of modern block ciphers, the evolutionary cryptanalysis on block ciphers grows greatly hard. To analyze quantitatively the validity of evolutionary cryptanalysis on block ciphers and the related issues, we design the fitness function by the ideas of maximum likelihood estimation and linear approximation, on which way we quantitatively analyze and experiment on two groups of examples that the key length is 49 and 42 bits respectively in six rounds of DES. The result indicates that the design of fitness function and the capability of resisting linearity approximation are the significant factors affecting the effects of evolutionary cryptanalysis on modern block ciphers. Compared with the reference [14], our experimental results improve significantly.

Our focus of study in the future is how to apply evolutionary cryptanalysis to block ciphers with higher intensity. Although there is rapid development in evolutionary cryptanalysis techniques, many issues need further study and exploration in this rising field.

Acknowledgments

The authors wish to thank the support by Excellent Young Teachers Program of China University of Geosciences (Grant. No.CUGQNL0820).

References

1. Spillman, R., Janssen, M., Nelson, B., Kepner, M.: Use of A Genetic Algorithm in the Cryptanalysis of simple substitution Ciphers. Cryptologia XVII(1), 187–201 (1993)
2. Forsyth, W.S., Safavi-Naini, R.: Automated cryptanalysis of substitution ciphers. Cryptologia 17(4), 407–418 (1993)
3. Clark, A.J.: Optimisation Heuristics for Cryptology [PhD thesis], Queensland. University of Technology (1998), http://sky.fit.qut.edu.au/~clarka/papers/thesis-ac.pdf

4. Grundlingh, W., van Vuuren, J.H.: Using Genetic Algorithms to Break a Simple Cryptographic Cipher (submitted, 2002) (Retrieved March 31, 2003), http://dip.sun.ac.za/~vuuren/abstracts/abstrgenetic.htm
5. Uddin, M.F., Youssef, A.M.: Cryptanalysis of Simple Substitution Ciphers Using Particle Swarm Optimization. In: Evolutionary Computation, CEC 2006, pp. 677–680 (2006)
6. Clark, A., Dawson, E.: A Parallel Genetic Algorithm for Cryptanalysis of the Polyalphabetic Subsitution Cipher. Cryptologia 21(2), 129–138 (1998)
7. Dimovski, D.G.: Alphabetic substitution cipher using a parallel genetic algorithm domain cooperation through SCOPES PROJECT, Ohrid, Maccedonia (2003)
8. Clark, A.: Modern optimisation algorithms for cryptanalysis. In: Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems, November 29, pp. 258–262 (1994)
9. Dimovski, A., Gligoroski, D.: Attacks On the Transposition Ciphers using Optimization Heuristics. In: Proceedings of ICEST 2003, Sofia, Bulgaria, pp. 1–4 (October 2003)
10. Russell, M., Clark, J.A., Stepney, S.: Making the most of Two Heuristics: Breaking Transposition Ciphers with Ants. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 146–147. Springer, Heidelberg (2003)
11. Toemeh, R., Arumugam, S.: Breaking Transposition Cipher with Genetic Algorithm. Electronics And Electrical Engineering 7(79), 75–78 (2007)
12. Nalini, N., Raghavendra Rao, G.: Cryptanalysis of Simplified Data Encryption Standard Via Optimaisation Heuristics. IJCSNS International Journal of Computer Science and Network Security 6(1B), 240–242 (2006)
13. Clark, J.A.: Invited Paper—Nature-Inspired Cryptography: Past, Present and Future, Conference on Evolutionary Computation. In: Special Session on Evolutionary Computation in Computer Security and Cryptography, CEC 2003, Canberra, December 2003, pp. 1647–1654 (2003)
14. Laskaria, E.C., Meletiouc, G.C., Stamatioud, Y.C., Vrahatisa, M.N.: Evolutionary computation based cryptanalysis: A first study. Nonlinear Analysis 63, 823–830 (2005)
15. Deng-guo, F.: The Design and Analysis of Block Cipher, p. 8. Tsinghua Press, BeiJing (2000)
16. Ming, Z., Shu-dong, S.: Genetic Algorithms: Theory and Application, p. 5. National Defense Industry Press, BeiJing (2002)
17. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991)

A K-Nearest-Neighbors Pareto Rank Assignment Strategy and Compound Crossover Operator Based NSGA-Ⅱ and Its Applications on Multi-objective Optimization Functions

Weiya Guo[1], Zhenhua Li[2], Dan Zhao[3], and Tim Wong[4]

[1] China University Of Geosciences
gaomin191025@163.com
[2] China University Of Geosciences
zhli@cug.edu.cn
[3] China University Of Geosciences
xinjiezhao@cug.edu.cn
[4] China University Of Geosciences
TIMWONG@263.net

Abstract. We try to improve the NSGA-II, one of the most classical MOP algorithms, in two ways. To measure individual crowding distance by edge weight of minimum spanning tree and k-nearest-neighbors Pareto rank assignment strategy is helpful on diversity of population; A compound crossover operator increases the extent and the ability of search. Experimental results on ZDTs and DTLZs, suggest that A K-Nearest-Neighbors Pareto Rank Assignment Strategy and Compound Crossover Operator Based NSGA-Ⅱ (KC NSGA-II) works faster and has more diverse solutions than its origins.

Keywords: Multi-Objective, Evolutionary Algorithm, NSGA-II, Minimum Spanning Tree, K-Nearest-Neighbors, Compound Crossover Operator.

1 Introduction

In reality, many problems can be attributed a number of grounds, may conflict with each other objective function of a multi-objective optimization problem. Over the years despite the multi-objective optimization problem has been solved in many ways, but the recent past 10 years, the evolutionary algorithm has been the ideal methods to solve the problem of multi-objective optimization, especially for solving the large-scale and complex multi-objective optimization problems, therefore the multi-objective optimization problem has become hot spots in the evolutionary algorithm research fields. The most advanced multi-objective Genetic Algorithms (NSGA-II) [1], [2], [3], in the world is a recently developed algorithm used to solve the multi-objective optimization problem[4], [5], [6], with a fine performance, which is widely used in all kinds of fields, and we also have had good results[6]-[14]. But because of its emerging, and there are many places to be improved, also, the research fields can not be comprehensive enough.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 142–151, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Therefore, how to combine the search strategy [16], and multi-objective optimization techniques[15], with the NSGA-II, will enable populations to Pareto front, and to the extent uniform across, which ultimately improve the quality of the solution, it is our an worthy issue of study. All of these studies will expand the application and research fields of the multi-objective optimization.

2 KC NSGA-II

2.1 Compound Crossover Operator

NSGA-II used the simulative binary crossover operator, the simulative binary crossover operator cross-operated to parent individuals by real-coded, to the given random crossover point, exchange the same side parts of intersection of two parent.

But simulative binary crossover operator used by NSGA-II is relatively weak performance of research and has deficiency of diversity of solution. In this paper, based on multi-father crossover operator [19], presents a compound crossover operator.

$$v_i = 2u_i - 1 \tag{1}$$

$$q_i = v_i \times |v_i|^c + 0.5 \tag{2}$$

u_i is a random number in [0, 1], in each dimension of two crossover individuals separately achieve different coefficient of non-convex combination, not as the individual a whole to non-convex combination, so as to accelerate the use of the current group effective information, which has a strong spatial search ability.

Record $x_1^t = (x_{11}^t, x_{12}^t, \ldots, x_{1n}^t)$ 、 $x_2^t = (x_{21}^t, x_{22}^t, \ldots, x_{2n}^t)$ as two individuals of the t generation, the formation process of these two individuals x_1、x_2 of the t+1th generation is as follows:

1. Randomly generate a random number $u_i \in [0, 1]$, i=1, 2, , n.
2. Use above crossover distribution function to calculate the linear combination coefficient q_i, obviously the range of q_i is[-0.5, 1.5].
3. Create two offspring individuals of the t+1th generation. Make the i-dimensional components x_{1i}^{t+1}、x_{2i}^{t+1}, then

$$x_{1i} = q_i \times x_{1i}^{t+1} + (1 - q_i) \times x_{2i}^{t+1} \tag{3}$$

$$x_{2i} = (1 - q_i) \times x_{1i}^{t+1} + q_i \times x_{2i}^{t+1} \tag{4}$$

Among them, c is crossover distribution index, its value is a real number that is beyond minus 1. When c is bigger, the probability of the next individual close to th parent is larger, and when c smaller, the probability of the next generation individual distance of parent is larger. Obviously, when the value of c is 0, q_i degenerates to obey the uniformly distribution in [-0.5, 1.5], the paper by several trials gets experience value is -0.75. What's more, firstly, rapidly sort the calculated q_i according to domain value and respond to quick sort individual i by fitness, then by them orthogonal multiply, finally get the crossover operator.

2.2 Crowding Operator

NSGA-II by calculating each evolutionary individual crowding distance[3] to estimate the density of individuals nearby. But this method has some shortcomings, such as some good distribution individuals may be eliminated, but poor distribution individuals may be left, to Fig 1:

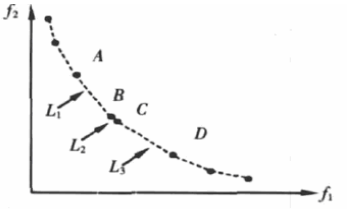


Fig. 1. Individual B and C may be eliminated or left at the same time

As shown in Fig1, the distance between individual B and C is very close, but from other individual is distant, according to the NSGA-II crowding distance, the crowding distance of B、C is closer to each other and larger than others, then B or C is likely to eliminate at the same time, or remained. This affected the distribution of the population, and the best result is selected one from B and C. Below with the minimum spanning tree [18], weight to each individual crowding distance can effectively solve the distance problem.

Firstly, calculate the Euclidean distance of each individual to the other N-1(N for population, or optimal surface individual number), and according to this create a minimum spanning tree that links all individuals. Now constitute an edge N non-circulating connected graph, each point(individual) crowding distance(congestion)is the length of an edge which is connected with others. What is worth noting is that each edge can only denotes ones, but the initial edge represents two.

In addition, adopting the boundary value reservation method, the best individual to a target, has directly maintained. KC NSGA-II has used the minimum spanning tree to express crowding distance, as in Figure 1, the individuals B、C crowding distances respectively are L1 and L2 or L2 and L3. So the distance of B、C is long, neither B and C will be eliminated at the same time, nor maintained. So those points that are in intensive region would not completely eliminated, and not be fully maintained.

2.3 Based on K- Nearest-Neighbors Pareto Rank Assignment Strategy

In NSGA-II, above crowding operating estimated to the density information, but the estimate is limited to individuals in the same non-dominated sets class. So, when a number of individuals with the same crowding distance, unable to determine which one is optimum. As shown in Figure 2, assumed that a as well as b has the same crowding distance, although the density of population around individual a is greater than the individual b surrounding population, their Pareto sort are the same 2, so the individual a and b still have the same opportunities for future generations manifold.

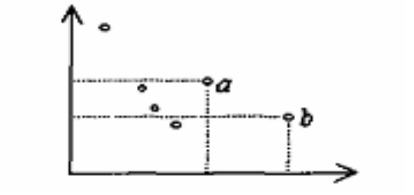


Fig. 2. Impact of population density around individuals

Based on above, take r(y, t) as the individual of t-th generation Pareto sort value, this strategy does following: First of all, similar to NSGA-II sorted all of the individuals by Pareto sort value, for every individual has a Pareto sort value. in the t-th generation population the k individuals dominate y are: y_1, y_2, …, y_k, the sort of individual y such as the definition of equation (5) below:

$$rank \quad (y,t) = 1 + \sum_{1}^{k} r(y_i, t) \tag{5}$$

Where, for individual a as well as b, their sort values are 4 and 2. The adopt strategy did not increase the complexity of NSGA-II, just do a little change to NSGA-II.

Thus, when some individuals calculated by the Minimum Spanning Tree have the same crowding distance, through the help of Pareto Rank-assignment Strategy based on Neighbors can be easy to choose the best individual. This assignment strategy has effectively enhanced the capacity of local search, and improved the convergence rate of solution to Pareto front.

3 Mutation Operator

KC NSGA-II has adopted Gaussian mutation operator: $N(0, \sigma_i)$, where the mean is 0, standard deviation isσ_i, the operator can be fine-tuning selected individual, which to some severe jitter functions is especially well. Its mutation probability following

$$P_m = a * \exp\left(-b\frac{gen}{\max\ gen}\right) \tag{6}$$

Gen represents evolution generation, maxgen is the greatest evolution generation, a and b are adjustable parameters, experience values for a = 0.05, b = 5. At the beginning and middle, the participations individual are more distant from Pareto front. Through using the greater probability, can maintain the diversity of population, and at the same time, increase the approximate rate of population. In the end the whole population has been basically close to the front, mutation rate rapidly decreases, so effectively prevent the population degenerating earlier.

4 The Algorithm

1. In whole space, randomly generate initial population P0 of size N
2. Sort populations with respect to its Pareto rank

3. Use selection, crossover and mutation to create a offspring populationQ_0 of size N
4. By new individuals of size N competed with the parent, get N individual members and with the compound crossover operation get temporary population Q_t
5. Sorting Q_t by non-dominated sorting, produce a series of non-dominated set F_t $=\{F_1, F_2, \ldots, F_m\}$, due to F_1 includes the best individuals of Q_t, add F1 into new parent population P_{t+1}, if the size of F_1 less than N, then continue add F_2 to P_{t+1}, until add F3, if the size of the population is beyond N, the individuals in F3 with crowding operation and considered whether to adopt the neighbors Pareto rank assignment strategy, reduce population of size N.
6. Make $P_t+1=Q_t$, $t=t+1$
7. When satisfying the termination conditions, stop, or else go to 3

5 Experiments

This paper has selected a series of ZDT and DTLZ to test, their Pareto solution sets are convex or non-convex, discontinuous or continuous, uniform and non-uniform, Test function can be found in references [3], [4], [5]. The evaluation performance of convergence and distribution are GD and SP [5].

5.1 ZDTs Test Functions and Experimental Results

We can notice from Table1, KC NSGA-II to solve the multi-objective function optimal front with convex or non-convex、discontinuous and convex, has performed better than NSGA-II, and is as well as the result of ZDT1-3 by MOIA-MSC in reference [21], But is more superior than ZDT4. Another, the obtained solutions compared to MST-NSGA-II in reference [19] is more precise, and most are better than SPEA2 and N-NOEA, of which mainly due to efficient crossover operator and based on K-Nearest-Neighbors Pareto rank assignment strategy. And from Figure 3、5、7、9、11 seen, KC NSGA-II has a good diversity and uniformity to non-dominated solution in the goal space, which was largely due to improved crowding operation and neighbor Pareto rank Strategy.

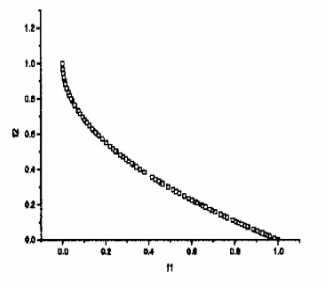


Fig. 3. Solution for ZDT1 by KC NSGA-II

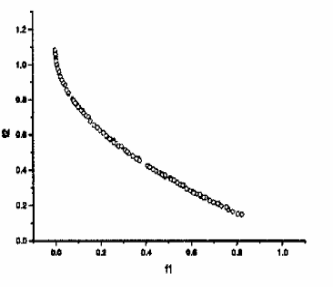


Fig. 4. Solution for ZDT1 by NSGA-II

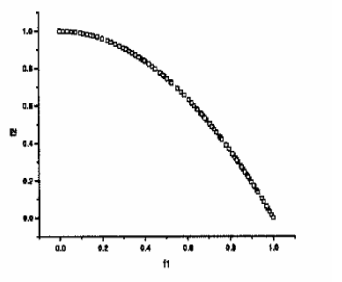


Fig. 5. Solution for ZDT2 by KC NSGA-II

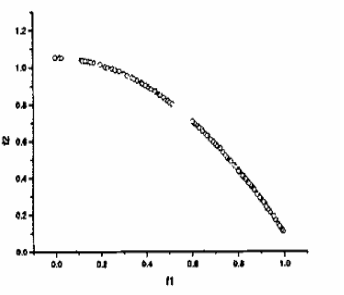


Fig. 6. Solution for ZDT2 by ZSGA-II

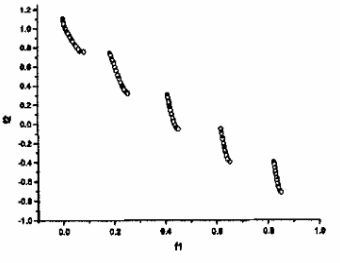


Fig. 7. Solution for ZDT3 by KC NSGA-II

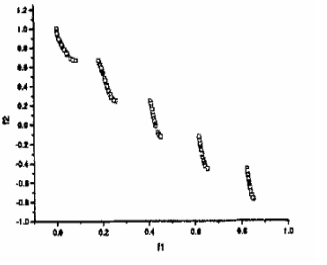


Fig. 8. Solution for ZDT3 by NSGA-II

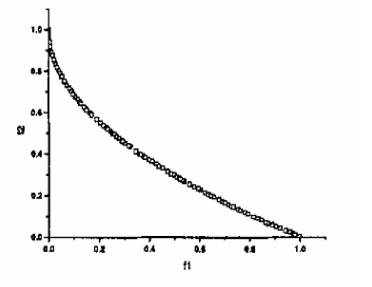


Fig. 9. Solution for ZDT4 by KC NSGA-II

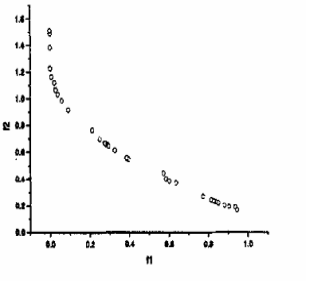


Fig. 10. Solution for ZDT4by NSGA-II

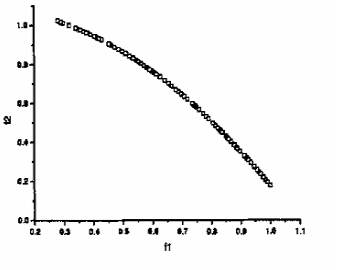


Fig. 11. Solution for ZDT6 by KC NSGA-II

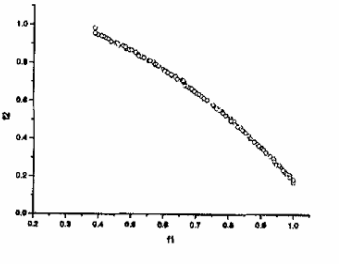


Fig. 12. Solution for ZDT6 by NSGA-II

Table 1. Performance comparison among some approaches

Algorithm		ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
KC NSGA-II	GD	**0.00016165**	**0.00009010**	**0.00050132**	**0.00018876**	**0.00050501**
	SP	0.00451000	0.00472610	0.00537099	**0.00406415**	**0.00141685**
	TIME/s	1.8775	0.8088	1.8117	**1.8200**	**1.8232**
NSGA-II	GD	0.00036451	0.00019301	0.00057591	0.00024440	0.05244596
	SP	0.00726419	0.00753598	0.00830991	0.00734001	0.00450008
	TIME/s	2.1188	2.1406	3.1002	1.8344	1.8312
MOIA-MCS	GD	0.00016170	0.00011735	0.00050135	0.01410210	0.00482019
	SP	0.00741549	0.0049136	0.01138468	0.02582847	0.00663389
	TIME/s	1.8875	1.7725	1.8118	1.9665	1.7300
MST-NSGA-II	GD	0.00020624	0.00010120	0.000556410	0.00018900	0.00050621
	SP	0.00471300	0.00482610	0.005375400	0.00484231	0.00351675
	TIME/s	2.0088	2.1506	3.1001	1.8300	1.8312
SPEA2	GD	0.00034259	0.00009142	0.00182973	0.06463929	0.00055196
	SP	**0.00317964**	**0.00296574**	0.01464783	0.00411846	0.0018444
	TIME/s	**24.4002**	**24.8565**	18.3023	20.1066	23.6722
N-MOEA	GD	0.00036046	0.00032841	0.00072975	0.00043619	0.00302169
	SP	0.0034302	0.0048987	**0.00440424**	0.00407644	0.00680886
	TIME/s	5.5656	4.1469	**5.3866**	4.9084	3.1535

5.2 DTLZ Test Function

Series of DTLZ test functions are put forward in 2005 [5] by Deb, which have a very important feature such as the target number is arbitrary scale variable. Here, choose a few examples to test. Fig13、14 are Pareto optimal front of DTLZ2.

5.2.1 DTLZ2 Test

From Fig13、14, KC NSGA-II found a better spread of DTLZ2 than NSGA-II.The distribution of NSGA-II optimal front is uneven, some region has no solution, some too many, KC NSGA-II is uniform, no apparent intensive region, and its distribution is more stable.

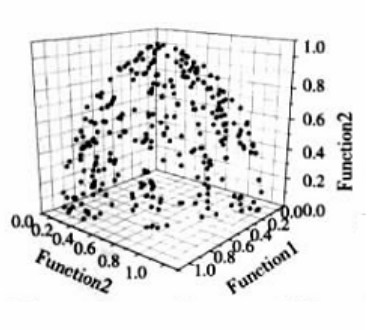


Fig. 13. Solutions of DTLZ2 by NSGA-II

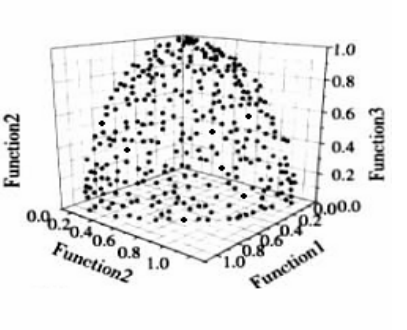


Fig. 14. Solution of DTLZ2 by KC NSGA-II

From Table2, it is noted that in both aspects of convergence and distribution of solutions KC NSGA-II performed better especially in low-dimensional than NSGAII.

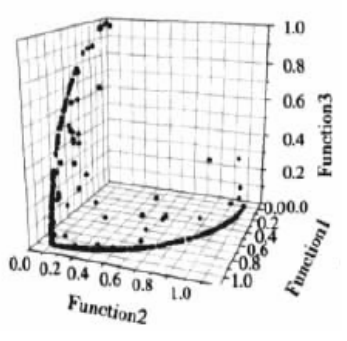
Table 2. Comparison of DTLZ2 for different objectives

Objective	GD		SP	
	NSGA-II	KC NSGA-II	NSGA-II	KC NSGA-II
2	0.0260110	0.0050047	0.013755	0.004838
3	0.0000467	0.0000165	0.034580	0.015547
4	0.0003269	0.0002345	0.070018	0.030886
6	0.0157390	0.0142360	0.201415	0.106515

DTLZ4 true Pareto front is non-uniform, in Function2=0 and Function3=0 curves it has a uniform line, and in other space it distributes evenly, as Figure 13、14 are the Pareto optimal font by two algorithms.

5.2.2 DTLZ4 Test Function

From Figure 15、16, the distribution of NSGA-II point is bad, some region is great density, some has emerged breakpoint; KC NSGA-II converged to the entire space, and the distribution of Function2 = 0, Function3 = 0 curves are better.



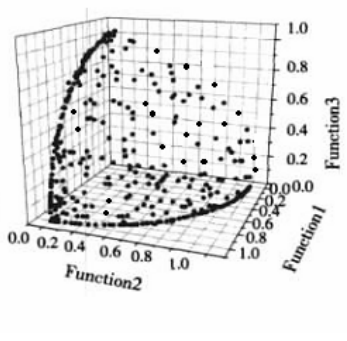


Fig. 15. Pareto curve of DTLZ4 by NSGA-II **Fig. 16.** Pareto curve of DTLZ4 by KC NSGA-II

Table 3. Comparison of DTLZ4 for different objectives

Objective	GD		SP	
	NSGA-II	KC NSGA-II	NSGA-II	KC NSGA-II
2	0.0005367	0.0002400	0.006941	0.006197
3	0.0010785	0.0003963	0.046568	0.033981
4	0.0006277	0.0005108	0.062464	0.047188
6	0.0042147	0.0021686	0.095962	0.069426

Table 3 has shown KC NSGA-II finds a better distribution of solution in DTLZ4 than NSGA-II, DTLZ4 is obvious in higher dimension, and in aspect of convergence NSGA-II performed worse than KC NSGA-II in low-dimensional.

6 Conclusion

1. NSGA-II's crowding operating estimation is limited to individuals of the same non-dominated sets class. We choose the best individual by calculating the Minimum Spanning Tree have the same crowding distance, with the help of Pareto Rank-assignment Strategy based on Neighbors. Experimental results show this strategy enhances local search ability, and improves the convergence rate of solutions.
2. Based on multi-parents crossover operator, we present a compound crossover operator. Experimental results have shown such a operator has a better global search capabilities, and maintain the diversity of the population well.
3. Experiments suggest that KC NSGA has better results of ZDTs and ZDTLs than NSGA-II by GD、SP, SPEA2、MST-NSGA-II、MOTA-MCS、N-MOEA, and also works faster, for example, it is 60-90% of NSGA's times, and 5% of SPEA2's times.

References

1. Srinivas, N., Deb, K.: Multi-objective optimization using non-dominated sorting ingenetic algorithms. Evolutionary Computation 2(3), 221–248 (1994)
2. Srinivas, N., Deb, K.: Multi-Objective Function Optimization Using Non-Dominated Sorting Genetical Algorithms. Evolutionary Computation 2(3), 221–248 (1995)
3. Deb, K.: Multi-objective genetic algorithms: Problem difficulties and construction of test functions. Evolutionary Computation 7(3), 205 (1999)
4. Mitra, K.: Multi-objective optimization of an industrial grinding operation using elitist nondominated sorting genetic algorithm. Chemical Engineering Science, 385–396 (2004)
5. Parsopoulos, K.E., Tasoulis, D.K., Pavlidis, N.G., Plagianakos, V.P., Vrahatis, M.N.: Vector Evaluated Differential Evolution for – Multiobjective Optimization
6. Liu, Y.: A fast optimization method of using nondominated sorting genetic algorithm (NSA-II) and 1-nearest neighbor (1NN) classifier for numerical model calibration. In: IEEE International Conference on 2005, pp. 544–549 (2005)
7. Akbari, A.A.: Pareto-Optimal Solutions for Multi-Objective Optimization of Turning Operation using Non-dominated Sorting Genetic Algorithm. In: TICME 2005, pp. 404–413 (2005)
8. Wei, X.: Ramp Shape Optimum Design for Airplane Land-Based Ski-Jump Take off via NSGA II. In: International Conference on 2006, pp. 995–1000 (2006)
9. Price, A.R.: Mul-tiobjective Tuning of Grid-Enabled Earth System Models Using a Non-dominated Sorting Genetic Algorithm (NSGA-II). In: IEEE International Conference on 2006, p. 117 (2006)
10. Nazemi, A.: Extracting a Set of Robust Pareto-Optimal Parameters for Hydrologic Models using NSGA-II and SCEM. In: IEEE Congress on 2006, pp. 1901–1908 (2006)
11. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test proiproblems. In: Congress on Evolutionary Computation, pp. 825–830 (2002)
12. Shi, Y.: A New Strategy for Parameter Estimation of Dynamic Differential Equations Based on NSGA II. Simulated evolution and learning, 345–352 (2006)
13. Bharti, S.: Optimal structural design of a morphing aircraft wing using parallel non-dominated sorting genetic algorithm II (NSGA II). Smart structures and materials, 616602-1–616602-12 (2006)

14. Xu, L.: Multi-objective Parameters Selection for SVM Classification Using NSGA-II. Advances in data mining, 365–376 (2006)
15. Goudos, S.K.: Electric Filter Optimal Design Suitable For Microwave Communications By Multi-Objective Evolutionary Algorithms. Microwave and Optical Technology Letters, 2324–2329 (2007)
16. Gallego, R.A., Monticelli, A., Romero, R.: Transmission system expansion planning by an extended genetic algorithm. IEE Proc. Gener,Transm. Distrib. 145(3), 329–335 (1998)
17. Guo, T., Kang, L.: A New Algorithm to Optimize Function with In equation and Restraints. Journal of Wuhan University: Natural Science 45(5), 771–775 (1999)
18. Huang, Z.: A Effective Multi-Objective Evolutionary Algorithm. Computer Engineering and Applications 43(11), 75–79 (2007)
19. Yu, R., Wang, Y.: A NSGA-II for Multi-Objective Minimum Spanning Tree Problem. Electronic Sci. & Tech. 126(6), 33–36 (2007)
20. Yang, S.: A Multi-Objective Evolutionary Algorithm Based on Parato Optimility and Limited Elitist. Computer Engineering and Applications 43(2), 108–111 (2007)
21. Peng, W., Huang, H.: A novel Multi-Objective Immune Algorithm Based on Memory Clonal Selection. Computer Engineering and Applications 44(16), 56–59 (2008)
22. Li, M., Zheng, J.: Mutil-objective evolutionary algorithm based on neighborhood. Computer Application 28(6), 1571–1574 (2008)

A Multi-objective Differential Evolutionary Algorithm Based on Spacial Distance

Jinhua Zheng, Jun Wu, and Hui Lv

Institute of Information Engineering, Xiangtan University, Hunan, China 411105

Abstract. Differential Evolution (DE) is a relatively new EA and it has become a major branch of EA. Based on the classical Differential Evolution, we proposed a new Multi-Objective Differential Evolutionary Algorithm in this paper. Compared with NSGA-II and ε-MOEA, the experimental results demonstrate that the new algorithm tends to be more effective in obtaining good convergence and can converge to the true Pareto front with comparable efficiency.

Keywords: multi-objective evolutionary algorithm, multi-objective optimal problem, Differential Evolutionary, Spacial Distance.

1 Introduction

Many real-world optimization problems usually involve multiple incommensurable and often competing objectives in nature, these problems are known as multi-objective optimization problems (MOOPs). The solution to a multi-objective optimization problem is made up of a set of solutions which we call Pareto solutions. EAs are characterised by the application of an in parallel search using a population of potential solutions. A multi-objective EA then incorporates the Pareto concept to identify multiple solutions through a single run of the algorithm.

Differential Evolution (DE) is a relatively new EA and it has been gaining popularity during previous years. It is type of single-objective evolutionary algorithm for optimization problems over a continuous domain. The main motivation of this approach is to adapt the search steps while the evolutionary algorithm is proceeding. It has been empirically shown that this simple strategy turns out to be effective in terms of convergence accuracy and speed. Several extensions of DE for multi-objective optimization have already been proposed. Some basic approaches just convert MOOPs to single-objective forms and use DE to solve these[1,2].

The first method extending DE for multi-objective optimization using the Pareto approach was Pareto-based DE approach[3]. Pareto Differential Evolution[4] was also mentioned at the same time, unfortunately without an explicit description of the method. After these, the Pareto(-frontier) Differential Evolution (PDE) algorithm[5] and a first version of Generalized Differential Evolution (GDE)[6] were introduced. The latest proposals were a second version of GDE[7] and The third version of GDE－GDE3[8], for global optimization with an arbitrary number of objectives and

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 152–161, 2008.
© Springer-Verlag Berlin Heidelberg 2008

constraints. In this paper, we present an improved MOEA based on the analysis of these algorithms, we call it SD-MODE. Our algorithm integrates the concept of spacial distance to maintain the diversity of the solutions and proposes an effective method to solve the problem of Cross-border for DE.

This paper is organized as follows: section 2 briefly introduces the differential concept and its operators; section 3 elaborates the main idea of the Multi-Objective Differential Evolutionary Algorithm Based on Spacial Distance; section 4 provides a briefing to the set of benchmark functions and the experimental results using the newly developed approach; and section 5 is the conclusion.

2 The DE Algorithm

The DE algorithm[9] was introduced by Storn and Price in 1995. Design principles in DE were simplicity, efficiency, and each variable's value in the chromosome is represented by a real number. Like a typical EA, DE has some random initial population, which is then improved using selection, mutation, and crossover operations. Several methods exist to determine a stopping criterion for EAs but usually a predefined upper limit for the number of generations or function evaluations to be computed provides an appropriate stopping condition.

In each generation DE goes through each decision vector $\vec{x}_{i,G}$ of the population and creates a corresponding trial vector $\vec{u}_{i,G}$. Here, i is an index of the vector in the population and G is a generation index. Creation of the trial vector is done as follows in the most common DE version, DE/rand/1/bin [10]:

$r_1, r_2, r_3 \in \{1,2,...,NP\}$, (randomly selected, except mutually different and different from i)

$j_{rand} = floor(rand_i[0,1) \times D) + 1$

For (j = 1; j <=D; j = j + 1)

{

if ($rand_j[0,1] < CR \vee j = j_{rand}$)

$u^i_{j,G} = x^{r3}_{j,G} + F \times (x^{r1}_{j,G} - x^{r2}_{j,G})$

Else

$u^i_{j,G} = x^i_{j,G}$

}

NP is the population size and D is the solution's dimension. The scaled difference between two randomly chosen vectors, $F \times (x^{r1}_{j,G} - x^{r2}_{j,G})$, defines magnitude and direction of the mutation. When the difference is added to a third randomly chosen vector $x^{r3}_{j,G}$, this corresponds to the mutation of the third vector.

Both CR and F are user defined control parameters for the DE algorithm and they remain fixed during the whole execution of the algorithm. Parameter CR, controlling the crossover operation, represents the probability that an element for the trial vector

is chosen from a linear combination of three randomly chosen vectors instead of from the old vector $x^{i}_{j,G}$. The condition "j = j_{rand}" is to make sure that at least one element is different compared to elements of the old vector. Parameter F is a scaling factor for mutation and its value is typically (0; 1]. In practice, CR controls the rotational invariance of the search, and its small value (e.g. 0.1) is practicable with separable problems while larger values (e.g. 0.9) are for non-separable problems. Control parameter F controls the speed and robustness of the search, i.e., a lower value for F increases the convergence rate but also the risk of stacking into a local optimum.

In all these common algorithms, there are two deficiencies: (1) not using an effective method to maintain the diversity of the solutions; (2) not using an effective method to keep the variables of solution in their boundaries in the process of crossover operation.

3 SD-MODE: A MOEA Based on Spacial Distance

A generic version of the adopted algorithm is presented in Figure 1. The SD-MODE algorithm is similar to the common DE with the following modifications:

1. It uses spacial distance to maintain the diversity of the solutions.
2. It uses an effective method to keep the variables of solution in their boundaries.
3. It uses recentness-optimal principle to generate the offsprings.

From the below described algorithm, we can see that there are two differences between common DE and SD-MODE:(1) In DE, a trial vector $u^{j}_{G=g}$ created by mutation and crossover operations is compared to an old vector $x^{j}_{G=g-1}$; But in our algorithm, we sort the parent population and child population by Pareto dominance after an evolutionary operation completed and then select NP individuals as parent population by spacial distance**;**(2) In DE, three parents are selected for crossover and the child is a perturbation of one of them, usually, it can be expressed by the formula: $u^{j}_{i,G=g} = x^{r3}_{i,G=g-1} + F \times (x^{r1}_{i,G=g-1} - x^{r2}_{i,G=g-1}) \quad if\ (rand[0,1) < CR \vee i = i_{rand})$, But in our algorithm, we use recentness-optimal principle to generate a trial vector $u^{j}_{G=g}$, two parents are selected for crossover and the child is a perturbation of the current vector $x^{j}_{i,G=g-1}$, it can be expressed by the formula: $u^{j}_{i,G=g} = x^{j}_{i,G=g-1} + F \times (x^{r1}_{i,G=g-1} - x^{r2}_{i,G=g-1}) \quad if\ (rand[0,1) < CR \vee i = i_{rand})$.

In the process of DE, we find that the trial vector may exceed its boundary, so we use a repair rule which can keep the trial vector stay in boundary, the repair rule is shown in expression(1)

$$u^{j}_{i,G=g} = \begin{cases} (x^{j}_{i,G=g-1} + u^{j}_{i,G=g})/2 & \text{while } u^{j}_{i,G=g} < \text{lower}(x_i) \\ upper(x_i) - (u^{j}_{i,G=g} - upper(x_i))/2 & \text{while } u^{j}_{i,G=g} > \text{upper}(x_i) \end{cases} \quad (1)$$

Algorithm 1

let G denotes a generation, P a population of size NP, $\overrightarrow{x^{j}_{G=k}}$ the j[th] individual of dimension D in population P in generation k and CR denotes the crossover probability, the size of mixpop is 2NP

input $D, NP \geq 4, CR \in [0,1], F \in (0,1]$ *and initial bounds:* $lower(x_i), upper(x_i), i = 1, ..., D$

initialize $P_{G=0} = \{\overrightarrow{x^{1}_{G=0}}, ..., \overrightarrow{x^{NP}_{G=0}}\}$ as

For each *individual* $j \in P_{G=0}$

$$x^{j}_{i,G=0} = lower(x_i) + rand_i[0,1] \times (upper(x_i) - lower(x_i)), \quad i = 1, ..., D$$

end for each

Evaluate $P_{G=0}$, *g=0;*

while *the stopping criterion is not satisfied* do

TempPop=$\varnothing$;

forall j≤NP

randomly select $r_1, r_2 \in \{1, 2, ..., NP\}$, *where* $j \neq r_1 \neq r_2$,

randomly select $i_{rand} \in (1, ..., D)$

forall i ≤ D, $u^{j}_{i,G=g}$ =

$$\begin{cases} x^{j}_{i,G=g-1} + F \times (x^{r1}_{i,G=g-1} - x^{r2}_{i,G=g-1}) \\ \quad if\ (rand[0,1) < CR \vee i = i_{rand}) \\ x^{j}_{i,G=g-1};\ otherwise \end{cases}$$

Repair $u^{j}_{i,G=g}$ *if any variable is outside its boundaries*

End **forall**

TempPop = TempPop$\bigcup u^{j}_{i,G=g}$

End **forall**

Newpop = P $\bigcup$ TempPop, *sort* Newpop *into sets of* F_1, F_2, ..., F_n *by Pareto rank*$\wedge$ Newpop =$F_1 \cup F_2 \cup \ldots \cup F_n$; P =$\Phi$,*i=1*; *if* $|F_1| >= NP$, *fill* P *with NP individuals from* F_I *by* **Algorithm 2**; *if* $|F_1| < M$, *then* $P_{g+1} = P_{g+1} \cup F_1$ *and fill* P *with* $NP - |F_1|$ *individuals from* Newpop$-F_1$ *by randomly; g=g+1;*

end while

return *the set* P *of non-dominated solutions.*

Fig. 1. The Multi-Objective Differential Evolutionary Algorithm Based on Spacial Distance

4 Experimental Results and Discussion

In order to test the performance of SD-MODE, we choose some representative and generally used benchmarks. They are: SCH[11], FON[12], ZDT1, ZDT2, ZDT3,

Algorithm 2

Step1: Calculate the Euclidean distance of any two individuals in F_i;

Step2: Select a pair of individuals with the shortest distance in the population, If there are more than two couples of individuals with the same inter-distance, select one couple randomly, we define the pair of individuals as (A, B). Afterwards, compare the each second shortest distance of A and B to others in the population, if the second shortest distance of A less than that of B, eliminates A from F_i, otherwise B;

Step3: If$| F_i|$> NP go to Step1, otherwise return F_{io}

Fig. 2. The archive truncation Algorithm

ZDT6[13]. The number of decision variables N: N_{SCH} =1, N_{FON} =2, N_{ZDT1} = N_{ZDT2} =$N_{ZDT3=}$ 30, N_{ZDT6} =10, all of them have 2 objectives. About the characteristic of these benchmarks, corresponding literatures can be referenced.

We use the well know NSGA-II and ε-MOEA as comparing algorithms. All MOEAs are given real valued decision variables. The parameter settings for them are set as that suggested in [14] and [15]. With η_c=15 for SBX, η_m=20 for polynomial mutation. The initial population size is set to 100 and the maximum number of generations to 200. Control parameters for SD-MODE is CR = 0.95. We find that for test function ZDT6 which is difficult converge, a large F (0.6) can get better value for the convergence and distribution, and for several other test functions, a smaller F (0.4) value is more suitable.

4.1 Performance Metrics

Several metrics have been proposed for measuring the results of Pareto-based multi-objective optimization algorithms. In this work we use the metrics *GD*[16] and *SP*[17]. The former was to measure the extent of spread achieved among the obtained solutions, and the latter measured the extent of convergence of known set of Pareto-optimal set.

GD measures the distance between the obtained nondominated front Q and the set P^* of Pareto-optimal solutions as

$$GD = \frac{\sqrt{\sum_{i=1}^{|Q|} d_i^2}}{|Q|} \tag{2}$$

where d_i is the Euclidean distance (in the objective space) between the solution $i \in Q$ and the nearest member of P^*. It must be pointed out that *GD* need to know the true Pareto front. Since we are dealing with artificial test problems, the true Pareto is not hard to obtain. In our experiments we use 500 uniformly spaced Pareto-optimal solutions as the approximation of the true Pareto front.

The second metric is *SP*. This metric measures the diversity of obtained solutions, $\bar{d}$ is the mean of all d_i, and M is the number of objectives.

$$SP = \sqrt{\frac{1}{|Q|-1}\sum_{i=1}^{|Q|}(\bar{d} - d_i)^2} \tag{3}$$

$$d_i = \min_{q_j \in Q \wedge q_j \neq q_i} \sum_{m=1}^{M} | f_m(q_i) - f_m(q_j) | \quad (4)$$

4.2 Experimental Results

For comparison, the SP and GD of the obtained solutions on all or some of the benchmarks by NSGA-II, ε-MOEA and SD-MODE are shown in Tables 1 and 2. For each problem, we have carried out 20 independent runs, and the table includes the average and standard deviation. Black body figures in the table denote that the value obtained by the corresponding algorithm is the smallest.

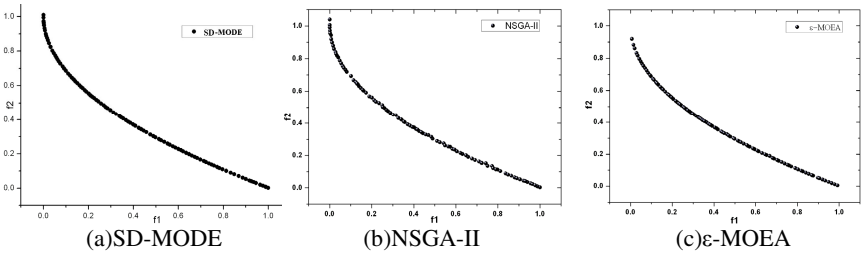
Table 1. Spacing Metric(SP) in 10 runs for NSGA-II, ε-MOEA and SD-MODE

MOEAs	SD-MODE		NSGA-II		ε-MOEA	
	SP (Avg)	SP(σ)	SP (Avg)	SP(σ)	SP (Avg)	SP(σ)
ZDT1	**0.003394**	0.000317	0.006029	0.000531	0.005904	**0.000280**
ZDT2	**0.003322**	**0.000297**	0.006159	0.000809	0.008615	0.000881
ZDT3	**0.004322**	**0.000513**	0.007135	0.000776	0.011164	0.000516
ZDT6	**0.002780**	**0.000669**	0.007925	0.003241	0.003638	0.000420
SCH	**0.014297**	**0.001398**	0.036418	0.002240	0.038069	0.003464
FON	**0.003612**	**0.000323**	0.008257	0.000506	0.022753	0.000399

Table 2. Convergence Metric(GD) in 20 runs for NSGA-II, ε-MOEA and SD-MODE

MOEAs	SD-MODE		NSGA-II		ε-MOEA	
	GD (Avg)	Std Dev	GD (Avg)	Std Dev	GD (Avg)	Std Dev
ZDT1	0.00034761	0.00014299	0.00040451	0.00028417	**0.00029911**	**0.00002844**
ZDT2	0.00027659	0.00007269	0.00037354	0.00028647	**0.00025649**	**0.00003717**
ZDT6	0.00106023	0.00055958	0.05244596	0.01802994	**0.00087653**	**0.00004602**
SCH	0.00037454	0.00001676	0.00042635	0.00000645	**0.00030888**	**0.00000282**

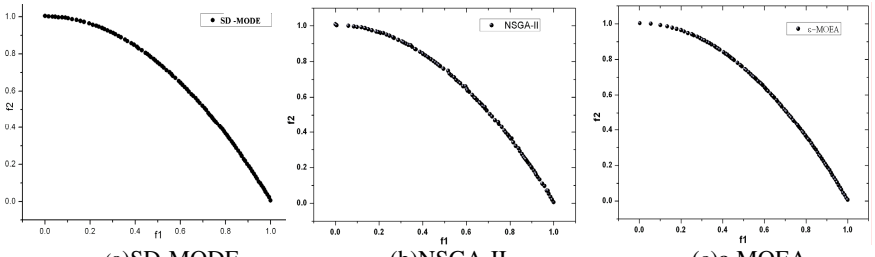
From Table 1, we know that for all problems, SD-MODE has the best performance in maintaining diversity of the solutions, ε-MOEA is worse than NSGA-II on ZDT2, ZDT3, SCH, FON, but it is better on ZDT1 and ZDT6. Table 2 told that ε-MOEA gains the best GD on all the four test problems. This is mainly because the ε-MOEA algorithm is essentially a grid-based method. Though SD-MODE is a little weaker than ε-MOEA in these problems on Convergence, it is also competitive with all of them. In the following, we'll give a more intuitionistic illumination by a set of figures.



(a)SD-MODE (b)NSGA-II (c)ε-MOEA

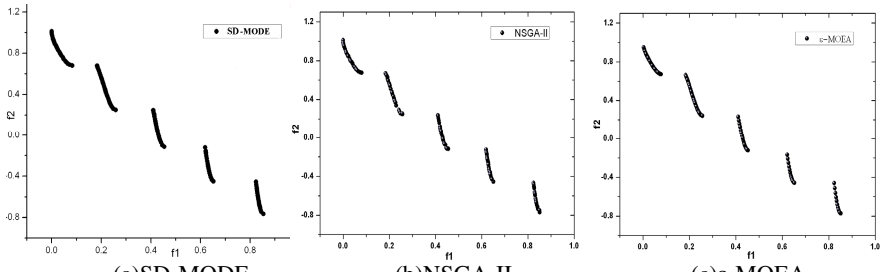
Fig. 3. PF of three MOEAs on ZDT1

Figure 3 and Figure 4 shows the similar situation of results. The PF obtained by NSGA-II is not so good and that by ε-MOEA is obviously sparser when the value of f2 approaching 1. The distribution of solutions obtained by SD-MODE is acceptable.



(a)SD-MODE (b)NSGA-II (c)ε-MOEA

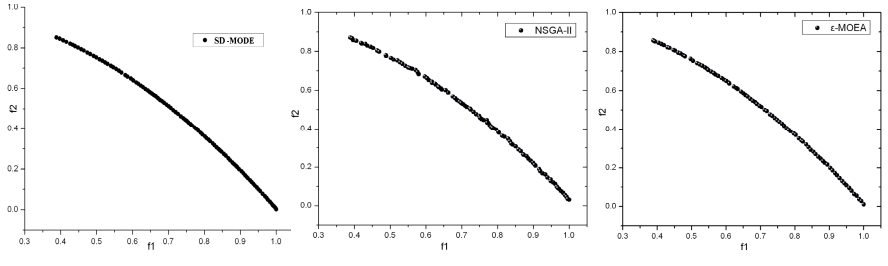
Fig. 4. PF of three MOEAs on ZDT2



(a)SD-MODE (b)NSGA-II (c)ε-MOEA

Fig. 5. PF of three MOEAs on ZDT3

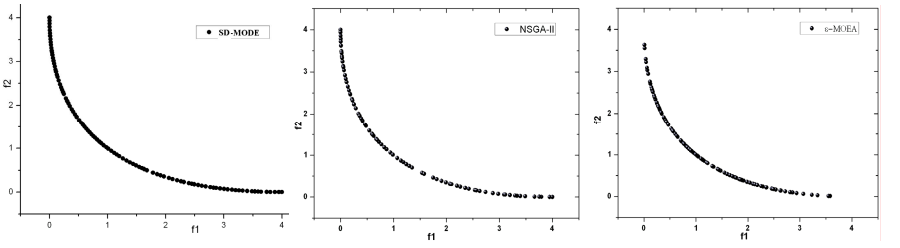
Figure 5 show that NSGA-II and ε-MOEA perform equally on diversity, but we can tell that SD-MODE own a more even solution set, and The PF obtained by ε-MOEA is obviously worse than that by the front two.



(a)SD-MODE (b)NSGA-II (c)ε-MOEA

Fig. 6. PF of three MOEAs on ZDT6

Figure 6 shows that The PF obtained by NSGA-II is not slippery and has blank region and that by ε-MOEA is not as good as that of SD-MODE. The distribution of solutions obtained by SD-MODE is very good.

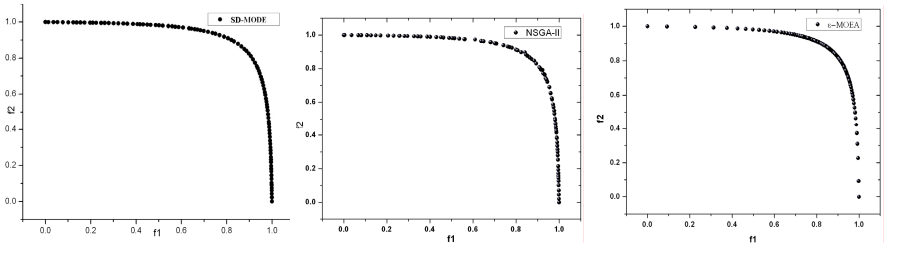


(a)SD-MODE (b)NSGA-II (c)ε-MOEA

Fig. 7. PF of three MOEAs on SCH

Figure 7 shows that the distribution of obtained solutions by NSGA-II seems discrete and that by ε-MOEA becomes sparser as going to the extremal regions of PF_{true}. Obviously SD-MOEA gained continuous and uniform solutions distributed on the PF_{true}.

Figure 8 illuminates the shortage of ε-MOEA clearly. It gets too many solutions crowded in the middle part but quite fewer ones in the extremal part. And NSGA-II has dissatisfying results while SD-MODE is still the best.



(a)SD-MODE (b)NSGA-II (c)ε-MOEA

Fig. 8. PF of three MOEAs on FON

From Figure 3-8, it can be seen that the Pareto Fronts obtained by SD-MODE, both in the breadth of distribution or in the aspect of uniformity, is better than that obtained by NSGA-II and ε-MOEA.

5 Conclusions and Future Research

This paper proposed SD-MODE based on the classical DE. There are two characteristics of the algorithm: (1) using spacial distance to maintenance the diversity so that the population maintained a good distribution, (2) using the recentness-optimal

principle to direct on the current vector with a disturbance. Comparing with NSGA-II and ε-MOEA, the results show that this algorithm have a better distribution and also a comparative convergence. How to make the algorithm has a good distribution of the solution set, simultaneously, an ideal convergence is an important goal of the future research work.

References

[1] Babu, B.V., Jehan, M.M.L.: Differential Evolution for multi-objective optimization. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003), Canberra, Australia, pp. 2696–2703 (December 2003)

[2] Chang, C.S., Xu, D.Y.: Differential Evolution based tuning of fuzzy automatic train operation for mass rapid transit system. IEE Proceedings on Electric Power Applications 147(3), 206–212 (2000)

[3] Chang, C.S., Xu, D.Y., Quek, H.B.: Pareto-optimal set based multi-objective tuning of fuzzy automatic train operation for mass transit system. IEE Proceedings on Electric Power Applications 146(5), 577–583 (1999)

[4] Bergey, P.K.: An agent enhanced intelligent spreadsheet solver for multi-criteria decision making. In: Proceedings of the Fifth Americas Conference on Information Systems (AMCIS 1999), Milwaukee, Wisconsin, USA, August 1999, pp. 966–968 (1999)

[5] Abbass, H.A., Sarker, R., Newton, C.: PDE:a Pareto-frontier Differential Evolution approach for multi-objective optimization problems. In: Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001), Seoul, South Korea, pp. 971–978 (2001)

[6] Lampinen, J.: DE's selection rule for multi-objective optimization. Technical report, Lappeenranta University of Technology, Department of Information Technology (2001) (April 1, 2005), http://www.it.lut.fi/kurssit/03-04/010778000/MODE.pdf

[7] Kukkonen, S., Lampinen, J.: An extension of Generalized Differential Evolution for multi-objective optimization with constraints. In: Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), Birmingham, Finland, pp. 752–761 (September 2004)

[8] Kukkonen, S., Lampinen, J.: Evolutionary Computation. In: The 2005 IEEE Congress on 2005, September 2-5, vol. 1, pp. 443–450 (2005)

[9] Storn, R., Price, K.V.: Differential Evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)

[10] Price, K.V.: New Ideas in Optimization. In: An Introduction to Differential Evolution, pp. 79–108. McGraw-Hill, London (1999)

[11] David, S.J.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: Grefenstette: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 93–100 (1985)

[12] Fonseca Carlos, M., Fleming, P.J.: An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation 3(1), 1–16 (1995)

[13] Kalyanmoy, D.: Multi-Objective Genetic Algorithms: Problem difficulties and Construction of Test Problems. Technical Report CI49/98. Department of Computer Science/LS11, University of Dortmund

[14] Kalyanmoy, D., Pratap, A., Agrawal, S., Meyrivan, T.: A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
[15] Deb, K., Mohan, M., Mishra, S.: A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. KanGAL Report. No 2003002
[16] Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary computation and convergence to a Pareto front. In: Late Breaking Papers at the Genetic Programming Conf., Madison, WI, pp. 221–228 (1998)
[17] Schott, J.R.: Fault tolerant design using single and multi-criteria genetic algorithm optimization. Master's theis of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge (1995-2005)

A Novel Opposition-Based Multi-objective Differential Evolution Algorithm for Multi-objective Optimization

Lei Peng[1,2], Yuanzhen Wang[1], and Guangming Dai[2]

[1] College of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China
[2] School of Computer, China University of Geosciences, Wuhan 430074, China
penglei0114@126.com

Abstract. Multiobjective optimization is of increasing importance in various fields and has very broad applications. The purpose of this paper is to describe a novel multiobjective optimization algorithm--opposition-based multi-objective differential evolution algorithm(OMODE). In the paper, OMODE uses the opposition-based population to generate the initial population of points, The important scaling factor is controlled by self-adaptive method. Performance of OMODE is demonstrated with a set of benchmark test functions and Earth-Mars double transfer problem. The results show that OMODE achieves better performance than other methods.

Keywords: opposition-Based, multi-objective optimization, Pareto-optimal solutions, differential evolution, OMODE, double transfer.

1 Introduction

Differential Evolution (DE), introduced by Rainer Storn[1] and Kenneth Price in 1995,is a branch of evolutionary algorithms for optimization problems over continuous domains. It has already been shown that DE is significantly faster and robust at numerical optimization and is more likely to find a function's true global optimum.

A few researchers[2][3][4][5][6]]have studied the extension of DE to solve multiobjective optimization problem in continuous domain. Abbass applied so-called Pareto Differential Evolution (PDE) algorithm to MOPs. Madavan got good results with the Pareto Differential Evolution(PDEA). Xue introduced Multiobjective Differential Evolution(MODE) and successfully used it solve enterprise planning problems. Tea proposed Differential Evolution for Multiobjective Optimization (DEMO) and achieved good results. In our previous studies[7], we proposed a self-adaptive multi-objective differential evolution(SMODE) to solve the spacecraft rendezvous problem.

This paper presents a novel opposition-based multi-objective differential evolution (OMODE) algorithm. It improves the efficiency of the SMODE over optimization of MOPs by applying opposition-based learning [8].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 162–170, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Opposition-Based Learning

The concept of opposition-based learning (OBL) was introduced by Tizhoosh [8] and has thus far been applied to accelerate reinforcement learning, backpropagation learning, and differential evolution[9]. The main idea behind OBL is the simultaneous consideration of an estimate and its corresponding opposite estimate (i.e. guess and opposite guess) in order to achieve a better approximation of the current candidate solution. Before concentrating on OBL, we need to define the concept of opposite numbers.

Definition 1. Let $P(x_1, x_2, ..., x_n)$ be a point in n-dimensional space, where $x_1, x_2, ..., x_n \in R$ and $x_i \in [a_i, b_i]$ $\forall i \in \{1, 2, ..., n\}$ .The opposite point of P is defined by $\breve{P}(\breve{x}_1, \breve{x}_2, ..., \breve{x}_n)$ where: $\breve{x}_i = a_i + b_i - x_i$. Now, by employing opposite point definition, the opposition-based optimization in MPOs can be defined as follows:

Opposition-Based Optimization. Let $P(x_1, x_2, ..., x_n)$ a point in an n-dimensional space with $x_i \in [a_i, b_i]$ $\forall i \in \{1, 2, ..., n\}$, be a candidate solution. Assume $f(x)$ is a fitness function which is used to measure candidate optimality. According to opposite point definition $\breve{P}(\breve{x}_1, \breve{x}_2, ..., \breve{x}_n)$ is the opposite of $P(x_1, x_2, ..., x_n)$. Now, if $\breve{P}$ dominates P , then point P can be replaced with $\breve{P}$; if P dominates $\breve{P}$, we continue with P ; otherwise, according to some probability, we decide which point continue.

3 The Opposition-Based Multi-objective Differential Evolution

In this section, the concept of opposition-based optimization is embedded in the SMODE[7] to enhance its capability for handling MOPs. At the same time, in order to extend simple DE to solve MOPs, we use the concept of dominance, nondominated sorting and crowding distance metric.

3.1 Opposition-Based Population Initialization

According to the literature[9],the opposition-based optimization can be utilized to accelerate searching methods since considering the pair P and $\breve{P}$ has apparently a higher fitness probability than pure randomness. So, We propose the following scheme:

(1) An individual is initialized with random according to the equation (1)

$$x_{ij}(0) = rnd(0,1)\left(U_{ij} - L_{ij}\right) + L_{ij}$$
$$i = 1, 2, \cdots, PopSize; \;\; j = 1, 2, \cdots, n \tag{1}$$

where PopSize is the population size, n is the solution's dimension, and $x_{ij}(0)$ is the j-th variable in the i-th solution vector, In the initial generation G = 0, $x_{ij}(0)$ is initialized within its boundaries (L_{ij} , U_{ij}).

(2) Calculating opposite individual $ox_{ij}(0)$, the i-th corresponding opposite individual for $ox_{ij}(0)$ is calculated by:

$$ox_{ij}(0) = L_{ij} + U_{ij} - x_{ij}(0) \tag{2}$$

(3) Selecting PopSize fittest individuals from the set $\{ x_{ij}(0), ox_{ij}(0) \}$ as initial population.

3.2 The Pseudo Code of OMODE

Based on the analysis above, the pseudo code of OMODE is described as follows:

/*Opposition-based population initialization*/
Random initialization of population X(i);
Calculate opposite population OX (i) ;
Selecting N fittest individuals from X(i) and OX (i) as initial population;
Evaluate initial population;
do /* Mutation and Crossover*/

$$u_{ij}(t) = \begin{cases} x_{r1j}(t) + F(x_{r2j}(t) - x_{r3j}(t)) & \text{if(} rand_{ij}[0,1] \le CR \text{ or } j = rand(i) \text{)} \\ x_{ij}(t) & \text{otherwise} \end{cases}$$

$F \in$ Gaussian$(0,1)$ is scaling factor, $CR \in [0,1]$ is the crossover factor.
If ($u_{ij}(t)$ outside their boundaries)

$$u_{ij}^{'}(t) = \begin{cases} L_{ij} + rand_{ij}[0,1](U_{ij} - L_{ij}) & if\,(u_{ij}(t) < L_{ij} \ or \ u_{ij}(t) > U_{ij}) \\ u_{ij}(t) & \text{otherwise} \end{cases}$$

/* Selection*/
If($u_{ij}^{'}(t)$ dominates $x_{ij}(t)$) $u_{ij}^{'}(t)$ replaces $x_{ij}(t)$
Else if ($x_{ij}(t)$ dominates $u_{ij}^{'}(t)$) $x_{ij}(t)$ replaces $u_{ij}^{'}(t)$
Else $u_{ij}^{'}(t)$ is added into the population
After that, the size of the new population is between PopSize and 2*PopSize, using nondominated sorting and crowding distance metric truncate it.
while the halting criterion is not satisfied

4 Experiments and Results

4.1 The Benchmark Functions

In order to test the performance of the OMODE, the five two-objective benchmark functions are used(ZDT1,ZDT2,ZDT3,ZDT4 and ZDT6[11]).

Two metrics was used. The smaller the value of these metrics, the better the performance of the algorithm. *Convergence metric* γ [10] measures the distance between

the obtained nondominated front Q and the set P* of Pareto-optimal solutions. *Diversity metric* Δ [10] measures the extent of spread achieved among the nondominated solutions.

In order to calculate the two metrics, we need to know the true Pareto front of the ZDT test problems. In this paper, we use 500 uniformly spaced Pareto-optimal solutions as the approximation of the true Pareto front.

The following parameters used for OMODE: Population size: NP = 100; Max generations: Max_gen = 250; Probability of crossover: CR = 0.3 or 0.5; Scaling factor: $F \in \text{Gaussian}(0,1)$;

Table 1-5 show the mean and variance of the values of the convergence metric and diversity metric obtained using NSGA- Ⅱ [10](real-coded and binary-coded), PAES[10], SPEA2[12], IBEA[12][13] and OMODE for ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, averaged over 10 runs,

Fig 1-5 present the nondominated fronts obtained by one run of OMODE.

Table 1. Mean and variances of the convergence and diversity metrics for ZDT1

Algorithm	ZDT1	
	Convergence Υ	Diversity Δ
NSAGA-II(real-code)	0.033482 ± 0.004750	0.390307 ± 0.001876
NSAGA-II(binary-code)	0.000894 ± 0.000000	0.463292 ± 0.041622
PAES	0.082085 ± 0.008679	1.229794 ± 0.004839
SPEA2	0.017186 ± 0.002197	0.509959 ± 0.041645
$\text{IBEA}_{\varepsilon+}$	0.006342 ± 0.000895	0.755664 ± 0.041579
IBEA_{HD}	0.006362 ± 0.001133	0.726860 ± 0.048051
OMODE	$\boldsymbol{0.001412 \pm 0.000240}$	$\boldsymbol{0.333536 \pm 0.025761}$

Table 2. Mean and variances of the convergence and diversity metrics for ZDT2

Algorithm	ZDT2	
	Convergence Υ	Diversity Δ
NSAGA-II(real-code)	0.072391 ± 0.031689	0.430776 ± 0.004721
NSAGA-II(binary-code)	0.000824 ± 0.000000	0.435112 ± 0.024607
PAES	0.126276 ± 0.036877	1.165942 ± 0.007682
SPEA2	0.025729 ± 0.003497	0.577083 ± 0.040160
$\text{IBEA}_{\varepsilon+}$	0.011880 ± 0.002973	0.916040 ± 0.029903
IBEA_{HD}	0.012789 ± 0.002039	0.895023 ± 0.056717
OMODE	$\boldsymbol{0.001100 \pm 0.000063}$	$\boldsymbol{0.310525 \pm 0.018051}$

Table 3. Mean and variances of the convergence and diversity metrics for ZDT3

Algorithm	ZDT3	
	Convergence Υ	Diversity Δ
NSAGA-II(real-code)	0.114500 ± 0.007940	0.738540 ± 0.019706
NSAGA-II(binary-code)	0.043411 ± 0.000042	0.575606 ± 0.005078
PAES	0.023872 ± 0.000010	0.789920 ± 0.001653
SPEA2	0.008087 ± 0.000976	0.692948 ± 0.030320
$\text{IBEA}_{\varepsilon+}$	0.003743 ± 0.000682	1.167969 ± 0.040211
IBEA_{HD}	0.004145 ± 0.000407	1.164363 ± 0.041839
OMODE	$\boldsymbol{0.001380 \pm 0.000073}$	$\boldsymbol{0.502986 \pm 0.026338}$

Table 4. Mean and variances of the convergence and diversity metrics for ZDT4

Algorithm	ZDT4	
	Convergence Υ	Diversity Δ
NSAGA-II(real-code)	0.513053 ± 0.118460	0.702612 ± 0.064648
NSAGA-II(binary-code)	3.227636 ± 7.307630	0.479475 ± 0.009841
PAES	0.854816 ± 0.527238	0.870458 ± 0.101399
SPEA2	0.706259 ± 0.235586	0.907895 ± 0.184215
$\text{IBEA}_{\varepsilon+}$	0.685179 ± 0.237895	0.955035 ± 0.038314
IBEA_{HD}	1.223929 ± 0.660444	1.067629 ± 0.029656
OMODE	$\boldsymbol{0.062202 \pm 0.083279}$	$\boldsymbol{0.396652 \pm 0.101442}$

Table 5. Mean and variances of the convergence and diversity metrics for ZDT6

Algorithm	ZDT6	
	Convergence Υ	Diversity Δ
NSAGA-II(real-code)	0.296564 ± 0.013135	0.668025 ± 0.009923
NSAGA-II(binary-code)	7.806798 ± 0.001667	0.644477 ± 0.035042
PAES	0.085469 ± 0.006664	1.153052 ± 0.003916
SPEA2	0.177565 ± 0.027220	0.825428 ± 0.047770
$\text{IBEA}_{\varepsilon+}$	0.106956 ± 0.019404	0.999535 ± 0.029938
IBEA_{HD}	0.125801 ± 0.019725	1.082538 ± 0.048338
OMODE	$\boldsymbol{0.005681 \pm 0.001019}$	$\boldsymbol{0.389825 \pm 0.205277}$

4.2 Earth-Mars Transfer Problem

Double impulse transferring is usually used in transferring from Earth to Mars. The primary aim[14] for double impulse transferring optimization is minimizing the fight time and launch energy. This problem is typical multi-objective optimization problem.

The two optimization objectives are flight time △t and launch energy (△V1+△V2).The optimization model is $\min\begin{cases} f_1=\Delta t \\ f_2=f(t_0,\Delta t)=\Delta v_1+\Delta v_2 \end{cases}$, The decision variables are the launch date,(t0 in MJD2000) and the mission time(△t in days). t0∈[800,3800] MJD2000,△t∈[100,400]days. OMODE and NSGA Ⅱ are used to solve this problem.

According to Fig.6, we can see that f_2 has some period, parameter t0 is divided into 4 sub-ranges. t0∈[800,1500], t0∈[1500,2200], t0∈[2200,3000] and t0∈[3000,3800]. Table 6 is the compare of two algorithms. The four typical solutions are chosen from the every ranges. From the Table 6,we can see that the majority solutions of OMODE are better than NSGA Ⅱ.

Table 6. Compare OMODE and NSGA Ⅱ in Earth-Mars transfer problem

range	OMODE			NSGA Ⅱ		
	t0	△t (days)	$\Delta V_1+\Delta V_2$ (km/s)	t0	△t (days)	$\Delta V_1+\Delta V_2$ (km/s)
[800, 1500]	1253.4205	203.3456	5.6687	1253.3239	203.7319	5.6686
	1254.7781	200.1344	5.6755	1257.0615	200.6993	5.6731
	1256.0209	185.3832	5.8340	1261.5621	186.1505	5.8401
	1258.9355	183.8274	5.8584	1258.2945	183.5183	5.8636
[1500, 2200]	2056.6811	215.2109	6.7882	2056.5495	215.2520	6.7881
	2057.4928	213.7925	6.7917	2057.307	214.3765	6.7901
	2058.2279	182.5630	7.4318	2063.9074	182.2305	7.4577
	2059.1303	180.7880	7.5004	2058.7167	180.2426	7.5212
[2200, 3000]	2812.6479	345.2185	6.1811	2812.5034	345.5998	6.1851
	2809.3835	331.6215	6.2791	2805.7006	331.25	6.2957
	2807.4960	322.1060	6.4823	2804.8684	317.0287	6.6352
	2827.7896	225.1438	7.6412	2834.4831	224.4116	7.6683
[3000, 3800]	3573.2686	324.3319	**5.6674**	3573.1108	324.1509	5.6695
	3573.0325	305.1108	5.8895	3574.4177	304.9983	5.8933
	3574.7590	299.7122	6.0836	3569.2432	294.4614	6.0431
	3597.7440	271.7664	6.8924	3597.52661	272.6796	6.8948

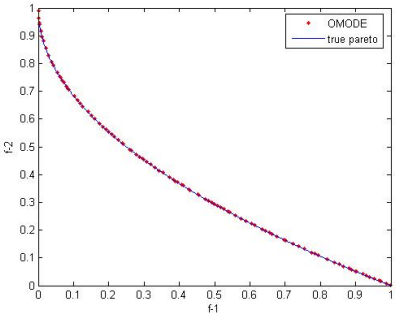


Fig. 1. Computed Pareto front on ZDT1

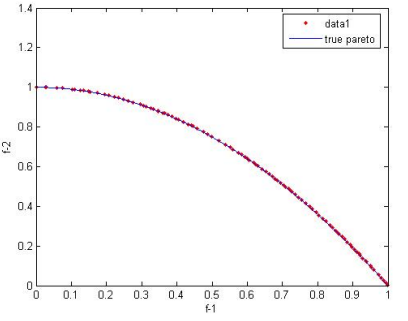


Fig. 2. Computed Pareto front on ZDT2

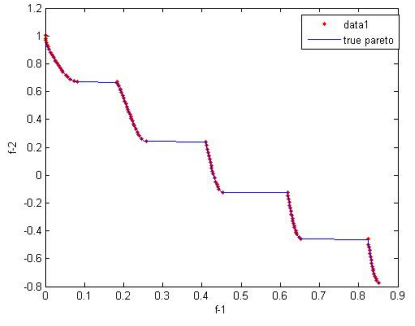


Fig. 3. Computed Pareto front on ZDT3

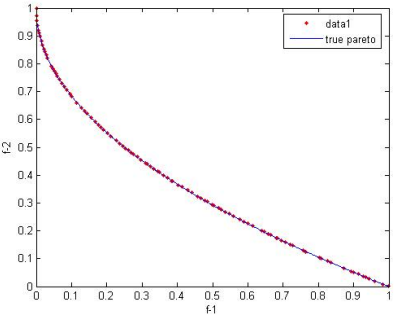


Fig. 4. Computed Pareto front on ZDT4

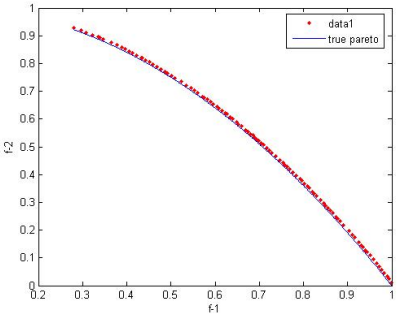


Fig. 5. Computed Pareto front on ZDT6

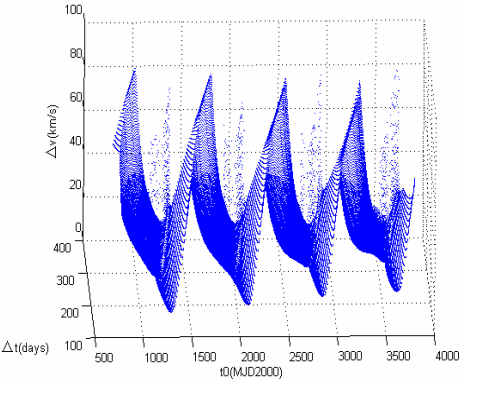


Fig. 6. The relationship between △t and △V1+△V2

5 Conclusion

In this paper, we propose an extension of DE algorithm based on opposition-based concept for MOPs. OMODE uses the opposition-based population to generate the initial population of points.

We test our proposed OMODE on a number of two objective benchmark problems and Earth-Mars transfer problem. From the analysis of the results, we can conclude that OMODE performs the best in these test problems than other compared algorithms to obtain a good distribution Pareto solutions on all of the test problems and better solution to Earth-Mars transfer problem than NSGA II.

In the experiments, we also find that CR=0.3 is better than CR=0.5 for ZDT1,ZDT2,ZDT3,ZDT6, but CR=0.5 is good to ZDT4. At the same time, for ZDT benchmark problems, DE/Rand/1/bin mutation scheme is better than others.

In the future work, some local search operators will be added. In the near future, OMODE will be used to solve the more multi-objective optimization problems in aerospace field.

Acknowledgements

The Project was supported by the Research Foundation for Outstanding Young Teachers, China University of Geosciences (Wuhan).

References

[1] Storn, R., Price, K.: Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)

[2] Abbass, H.A.: The self-adaptive pareto differential evolution algorithm. In: Congress on Evolutionary Computation (CEC 2002), Piscataway, New Jersey, vol. 1, pp. 831–836. IEEE Service Center, Los Alamitos (2002)

[3] Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003), Canberra, Australia, vol. 2, pp. 862–869. IEEE Press, Los Alamitos (2003)

[4] Robič, T., Filipič, B.: DEMO: Differential Evolution for Multi-objective Optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 520–533. Springer, Heidelberg (2005)

[5] Babu, B.V., Mathew Leenus Jehan, M.: Differential Evolution for Multi-Objective Optimization. In: CEC 2003, Canberra, Australia, vol. 4, pp. 2696–2703 (December 2003)

[6] Madavan, N.K.: Multiobjective optimization using a Pareto differential evolution approach. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 1145–1150 (2002)

[7] Peng, L., Dai, G., Chen, F., Liu, F.: Study on Application of Multi-Objective Differential Evolution Algorithm in Space Rendezvous. In: Kang, L., Liu, Y., Zeng, S. (eds.) ISICA 2007. LNCS, vol. 4683, pp. 46–52. Springer, Heidelberg (2007)

[8] Tizhoosh, H.R.: Opposition-based learning: A new scheme for machine intelligence. In: Proc. Int. Conf. Comput. Intell. Modeling Control and Autom., Vienna, Austria. 2005, vol. I, pp. 695–701 (2005)

[9] Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-Based Differential Evolution Algorithms. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 2010–2017 (2006)
[10] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGACII. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
[11] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8, 173–195 (2000)
[12] Zhang, J.: Research on Indicator-Based Evolutionary Algorithm and Its Application in Constellation Design.Master degree thesis. China University of Geosciences, Wuhan, China (2008)
[13] Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) 8th Int'l. Conf. on Parallel Problem Solving from Nature (PPSN VIII), UK, pp. 832–842. Springer, Heidelberg (2004)
[14] Myatt, D.R., Becerra, V.M., Nasuto, S.J., et al.: Advanced Global Optimization for Mission Analysis and Design, pp. 33–37, http://www.esa.in/act

Acquainted Non-convexity Multiresolution Based Optimization for Affine Parameter Estimation in Image Registration

J. Dinesh Peter, V.K. Govindan, and Abraham T. Mathew

National Institute of Technology Calicut - India

Abstract. Affine parameter estimation technique applied to image registration is found useful in obtaining reliable fusion of same object's images taken from different modalities, into single image with strong features. Usually, the minimization in affine parameter estimation technique can be done by least squares in a quadratic way. However, this will be sensitive to the presence of outliers. Therefore, affine parameter estimation technique for image registration calls for methods that are robust enough to withstand the influence of outliers. Progressively, some robust estimation techniques demanding non-quadratic and non-convex potentials adopted from statistical literature have been used for solving these. Addressing the minimization of error function in a factual framework for finding the global optimal solution, the minimization can begin with the convex estimator at the coarser level and gradually introduce non-convexity i.e., from soft to hard redescending non-convex estimators when the iteration reaches finer level of multiresolution pyramid. Comparison has been made to find the performance results of proposed method with the registration results found using different robust estimators.

Keywords: Image Registration, Motion estimation, Affine parameter estimation, outliers, Robust M-estimators.

1 Introduction

Image registration is the process of metamorphosing different sets of information present in an image into another coordinate system. It is also used to compare or integrate the information from two or more images obtained from different measurements or modalities. Among many registration algorithms which contends with both large overall transformations and with more localized non-rigid transformations is differential affine motion estimation based registration technique [1-3]. This paper has discussed various constraints like smoothness [4] and radiometric constraints [5-6] which make the algorithm to estimate the changes in overall visual aspect and large intensity variations, producing effective registration for the images taken from both inter and intra imaging modalities. Moreover, these constraints are much useful for one to estimate the exact affine

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 171–180, 2008.
© Springer-Verlag Berlin Heidelberg 2008

parameters by finding the solution of the energy function with globally optimum values. Two frequently used techniques for affine parameter estimation in computer vision are least squares and M-estimator based techniques. Least square technique together with geometric transformation can be seen in [7-8]. This approach can handle the outliers to some extent, but has low statistical efficiency. Amongst the methods suggested in the context of robust statistics, M-estimators lead a compromise between algorithmic, computational complexity, outlier rejection potentiality and high statistical efficiency. A novel factual framework for minimization with the help of multiresolution strategy and robust statistics can be applicable for all the transformation cases is proposed. Various robust M-estimators are selected in the order from their property of convexity to non-convexity. Following the idea of Graduated Non-convexity (GNC) algorithm described in [9], non-convexity is gradually acquainted from convexity by using the chosen robust M-estimators when the multiresolution pyramid scheme reaches its finer resolution. This results in a new and non-monitored robust affine parameter estimation for image registration. In this paper, the similarity measure between two images is sum of squared difference (SSD). In the case of two images from same imaging modality, this similarity measure is optimal. In [15], the case where the intensity transformation is not-known or different between two images (i.e., two images are taken from different modalities), various similarity measures were compared.

The rest of the paper is organized as follows. Section 2 explains the proposed methodology followed for the alignment of images. Section 3 compares the image registration accuracies of the proposed scheme with various estimators applied individually in affine parameter estimation technique. Section 4 summarizes the yardsticks and the results.

2 Methodology

2.1 Robust M-Estimation Function Selection

Estimation function is the function of intensities. The properties that are to be satisfied for the estimation functions in parameter estimation problem have been defined in various works in the literature. This paper describes the use of three different robust M-estimators in the parameter estimation problem for image registration[1]. Among those, one is convex potential and the other two are non-convex potentials (see Table.1). Further claim of the selection of these robust M-estimation functions are explained in section 2.2.

The chosen convex potential is *Charbonnier* objective function (E_{CH}) [16]. Since *Charbonnier* function (E_{CH}) is continuous, strictly increasing and differentiability is desirable for ψ, the solution to the minimization problem always exists and there is a guarantee of minimization up to global minimum [10]. Charbonnier function (E_{CH}) also satisfies the following conditions suggested by [16].

[1] For 128×128 input images, there will be three levels of resolution in multiresolution pyramid scheme.

Table 1. Robust objective functions and its properties

Property	**Robust function**	**Objective function** $\Re(x)$	**Influence function** $\psi(x)$
Convex	Charbonnier (E_{CH})	$2\sqrt{1+(\frac{x}{\sigma})^2}-2$	Monotone
Non-convex	Hebert-Leahy (E_{HL})	$\log(1+(\frac{x}{\sigma})^2)$	Soft Redescender
Non-convex	Tukey's Biweight (E_T)	$\begin{pmatrix} \frac{x^2}{\sigma^2}-\frac{x^4}{\sigma^4}+\frac{x^6}{3\sigma^6} & \mid x \mid \leq \sigma \\ \frac{1}{3} & otherwise \end{pmatrix}$	Hard Redescender

* E_{CH} is continuous and strictly decreasing on $[0\ +\infty]$ (See Fig.1).
* $\underset{(\frac{x}{\sigma}) \to +\infty}{Lim} E_{CH} = 0$ (For larger distance (Outliers), Weights are Zero).
* $\underset{(\frac{x}{\sigma}) \to 0+}{Lim} E_{CH} = W,\ 0 < W < +\infty$ (From smaller to medium values, considerable weights are provided.)

The other chosen non-convex potentials are Hebert-Leahy (E_{HL}) [11] and Tukey's biweight function (E_T). To ensure the goal of robustness, a redescending norm (Influence function or ψ-function) is required. From this point of view, hard redescending norm such as Tukey's biweight (E_T) is more attractive to outliers. Fig.1 shows the influence functions (ψ-functions) of the various robust objective functions given in Table.1.

Hard redescending norms yield high non-convex potentials, which lead the minimization steps up to local minimum only. So an efficient factual framework can however be determined in this instance for reducing the influence of outliers as well as the minimization up to global minimum. The method proposed in the next section will explain the same idea for estimating the exact affine parameters.

2.2 Acquainted Non-convexity Multiresolution Based Minimization Scheme (ANMMS)

A coarse to fine strategy is followed in this case. The Gaussian pyramid is built by convolving the original image with a set of gaussian weight functions and then

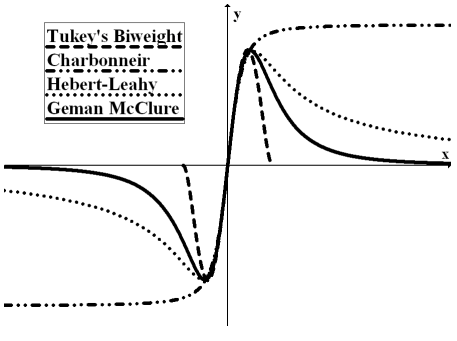


Fig. 1. (a)Influence functions (ψ-functions) of the chosen M-Estimators shown in Table. 1

sub-sampling it. Depending upon the normal distribution of the gaussian weight functions used for pyramid generation, non-convex M-estimators can be chosen from their property of soft redescending norm to hard redescending norm. During the middle level of resolution in multiresolution scheme, the impact of outliers is less. So choosing soft redescending norm at this stage improves the parameter estimation by finding optimally minimum values. And during the finer resolution level, the impact of outliers is more, so hard redescending norm is required to reduce the influence of outliers dramatically. The parameters estimated at the finer resolution will be used to warp the original source to produce the registered image. Depending upon the number of levels of resolution in multiresolution pyramid approach, different robust norms can be selected for operation. Fig. 2 shows the overview of the proposed scheme for minimization to estimate the accurate affine parameters and its application to image registration. In this, 128×128 images are initially padded with zeroes to become 160×160 images in order to handle the edge consequences. Then a three-level Gaussian pyramid (40×40, 80×80 and 160×160 images) is constructed for source and target images using 5-tap low-pass filter. Our implementation uses iterative re-weighted least squares ($IRLS$) method at each level of resolution. Once the final affine parameters $\boldsymbol{P}$ have been estimated, the original source image can be warped so as to produce a registered image using some standard re-sampling techniques [14]. This proposed methodology obviously reduces the complexity when compared with the Graduated Non-convexity algorithm proposed by Black and Zisserman [9] and its application to parameter estimation in [7].

2.3 Robust Gradient-Based Image Registration

The images for image registration are represented as $I_{x,y,t}$ (Source Image) and $I_{x,y,t-1}$ (Target Image)[2]. The basic flow of the algorithm follows the principles carried on [7]. For the spatial neighborhood Ξ of the source and target images, the error function can be written as:

$$E(\boldsymbol{P}) = \sum_{x,y \in \Xi} \Big[I_{x,y,t} - I_{P_1 x + P_2 y + P_5, P_3 x + P_4 y + P_6, t-1} \Big]^2 \tag{1}$$

where $\boldsymbol{P} = (\boldsymbol{P}_1, \cdots, \boldsymbol{P}_6)^T$ represents the affine parameters. In order to give attention to the outliers in an image, a robust objective function is applied in the error function. Since the error function is nonlinear in its unknown affine parameters, it cannot be minimized analytically. In order to simplify the minimization, first order truncated Taylor series approximation is applied here.

$$E(\boldsymbol{P}) \approx \sum_{x,y \in \Xi} \Re \Big\{ \Big[I^t_{x,y,t} - (P_1 x + P_2 y + P_5 - x) I^x_{x,y,t} - (P_3 x + P_4 y + P_6 - y) I^y_{x,y,t} \Big]^2, \sigma \Big\} \tag{2}$$

[2] These unconventional notations represent the source and target image with the temporal parameter t. These notations have been used only for mathematical consistency.

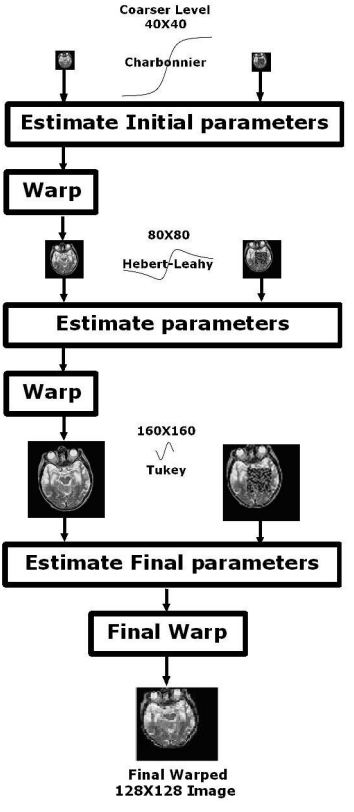


Fig. 2. Overview of image registration using acquainted non-convexity multiresolution based minimization scheme for 128×128 images

where $I^x_{x,y,t}$ and $I^y_{x,y,t}$ are spatial derivatives and $I^t_{x,y,t}$ is temporal derivative. $\Re$ represents the usage of robust M-estimator in the error function. The expression considered so far does not yield attention to the image constituent such as intensity variations over time. The introduced robust framework can be robust to the intensity variations by treating those as outliers. So the radiometric parameters are introduced to mind the intensity variations in the following style.

$$c(I_{x,y,t}) + b = I_{P_1x+P_2y+P_5, P_3x+P_4y+P_6, t-1} \tag{3}$$

c and b are contrast and brightness parameters (also called radiometric parameters) respectively. Now the error function can be represented in a simpler form like

$$\begin{aligned} E_{cb}(\boldsymbol{P}) \approx \sum_{x,y \in \Xi} \Re\Big\{ \Big[I^t_{x,y,t} - (P_1x + P_2y + P_5 - x) I^x_{x,y,t} - \\ (P_3x + P_4y + P_6 - y) I^y_{x,y,t} - \\ (I_{P_1x+P_2y+P_5, P_3x+P_4y+P_6, t-1} - (P_7 I_{x,y,t} + P_8)) \Big]^2, \sigma \Big\} \end{aligned} \tag{4}$$

After the addition of radiometric parameters to the error function, eight parameters $\boldsymbol{P} = (\boldsymbol{P}_1, \cdots, \boldsymbol{P}_8)^T$ are estimated. Here $\boldsymbol{P}_7$ and $\boldsymbol{P}_8$ are contrast and brightness parameters respectively. And now the reduced form of the error function can be written as

$$E_{cb}(\boldsymbol{P}) \approx \sum_{x,y \in \Xi} \Re\Big\{ \Big[r - \boldsymbol{q}^T \boldsymbol{P} \Big]^2, \sigma_{cb} \Big\} \tag{5}$$

where the vector $\boldsymbol{q} = \Big(xI^x \;\; yI^x \;\; xI^y \;\; yI^y \;\; I^x \;\; I^y \;\; -I \;\; -1 \Big)^T$ and the scalar $r = \Big(I^t + xI^x + yI^y - I \Big)$. This notation has been used for removing the complexity of understanding the expressions. It can be presumed that the motion of image surface vary gradually. Thus imposing smoothness constraints on the motion of adjacent points in the image plane replaces the intensity gradient constancy assumption along motion trajectories. So the error on the smoothness assumption E_s can be defined as

$$E_s(\boldsymbol{P}) = \sum_{x,y \in \Xi} \left(\vartheta^T \left(\Big(\frac{\partial \boldsymbol{P}}{\partial x} \Big)^2 + \Big(\frac{\partial \boldsymbol{P}}{\partial y} \Big)^2 \right) \right) \tag{6}$$

$\vartheta^T \in Z^+$contains eight constant values for each parameters $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_6 \boldsymbol{P}_7 \boldsymbol{P}_8$ that are used to control the smoothness constraint. Now the augmented robust error function is as follows

$$E(\boldsymbol{P}) = \Re\Big\{ E_{cb}(\boldsymbol{P}) + E_s(\boldsymbol{P}), \sigma \Big\} \tag{7}$$

Perhaps smoothness constraint E_s, radiometric constraints (c, b) have the added benefit of preventing weak solution in the robust framework. This error function can now be minimized analytically by differentiating the error function with respect to its unknown affine parameters. The derivative of $E_{cb}(\boldsymbol{P})$ and $E_s(\boldsymbol{P})$ are as follows

$$\frac{dE(\boldsymbol{P})}{d\boldsymbol{P}} = \sum_{x,y \in \Xi} \psi \Bigg\{ \left(\Big[r - \boldsymbol{q}^T \boldsymbol{P} \Big]^2 + \vartheta^T \left[\Big(\frac{\partial \boldsymbol{P}}{\partial x} \Big)^2 + \Big(\frac{\partial \boldsymbol{P}}{\partial y} \Big)^2 \right] \right), \sigma \Bigg\} \tag{8}$$

Here $\psi(x, \sigma) = \Re'(x, \sigma)$ represents the robust influence function. Bringing the model into the gradient constraint (spatial/temporal derivatives) and considering all the pixels in the image at the same time, a linear influence system is solved by differentiating the error function with respect to its unknown affine parameters. Now the iteration scheme starts by involving robust weight function $W\Big(\frac{Fe_i}{\sigma}\Big)$ for parameter estimation. An exact estimate of the error function during minimization can be regulated using Iterative re-weighted least squares technique (IRLS), where the weights are being modified on each iteration. The estimated parameters are fine-tuned on each iteration as follows

$$\boldsymbol{P}_{j+1} = \Bigg[\sum_{x,y \in \Xi} \Big(\boldsymbol{q} * W\Big(\frac{Fe_i}{\sigma} \Big) * \boldsymbol{q}^T \Big) + \Phi \Bigg]^{-1} \Bigg[\sum_{x,y \in \Xi} \Big(\boldsymbol{q} * W\Big(\frac{Fe_i}{\sigma} \Big) * r \Big) + \Phi \int \boldsymbol{P}_j \Bigg] \tag{9}$$

where $\boldsymbol{P}_j$ are the estimated parameters at iteration j. W is a diagonal matrix whose elements are weights of the chosen robust weight function and the residual $Fe_i = r - \boldsymbol{q}^T \boldsymbol{P}$. $\int \boldsymbol{P}$ is the average of $\boldsymbol{P}$ over a small neighborhood window in an image which enforces smoothing. Φ is now the diagonal matrix with the diagonal elements of ϑ^T. At every level, the scale parameter σ for the weight function is calculated by using residuals of the error function. The equation for finding σ is given in [17] as

$$\sigma = 1.48\, median\left\{\left|Fe_i - median\{Fe_i\}\right|\right\} \tag{10}$$

The iteration stops when the total squared error in each iteration do not show a substantial step-down with respect to the error in the previous iteration, or whenever it reaches maximum number of iterations. In each iteration, the error function is calculated as

$$E^2 = \frac{\sum_{i=1}^{I(N)} W_i Fe_i^2}{\sum_{i=1}^{I(N)} W_i} \tag{11}$$

And now, the parameters at each level of resolution are used to warp the source image in the next level of multiresolution pyramid using bi-cubic interpolation method. This process is repeated at every level of resolution. The transformation at all levels of multiresolution pyramid is accrued and concedes a single transformation. Now enforcing this final transformation to the original source acquires the registered image. For computing spatial/temporal derivatives, this method utilizes some set of derivative filters [12]. We have worked only with 2D MRI brain atlas images and the results are shown for the same.

3 Results and Discussion

Source and target images are 128×128, 8-bit gray-scale images. Some irrelevant random information has been added in the target image that can act as outliers. This will impose absolutely wrong parameter estimation while using least square approach for the minimization scheme. The proposed method deals with an approach based on robust M-estimators to reduce the influence of outliers. The images used for experimenting are brain atlas images that are taken from *Harvard medical university website* [13]. Matlab implementation of this method requires approximately 2 minutes for 128×128 images on Pentium IV, 3 GHz, 512 MB RAM in Windows operating system.

Fig.3 shows the registration result for various rotated versions of source image with the added outliers. This shows the root mean square error value for each registered image with the original source image. Efficient results have been found while using *ANMMS* approach. Almost constant RMS value are shown for the registration of various rotated versions of source image. The other results are obtained by using individual non-convex robust estimator applied in all the levels of resolution in multiresolution strategy.

Fig.4 (a) and (b) shows the estimated translation parameters in X-axis as well as Y-axis respectively. The results are shown for the proposed scheme as well

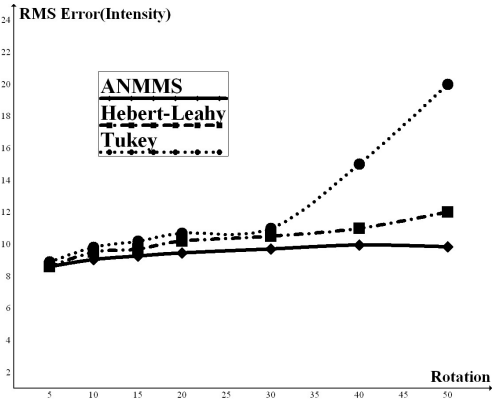


Fig. 3. RMS Error between registered images of various rotated versions of source image with original source image

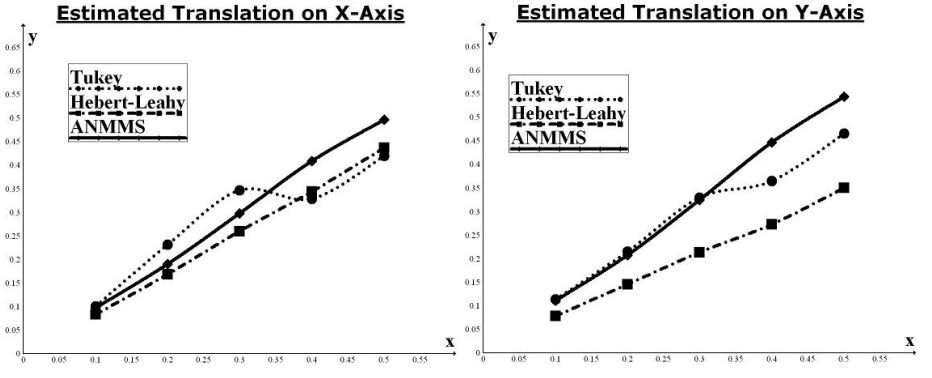


Fig. 4. (a) Estimated Translation parameters of registered source images for different translated versions of source image in X-axis (b) same with Y-axis

as for the individually applied robust estimators in the multiresolution pyramid. These figures show that the proposed scheme provides the estimated translation parameters that are linearly estimated towards the translated version of source images.

Table 2. Mean (and Standard Deviation) of estimated registration parameters

Robust Methods	T_x	T_y	K_y	θ_x
Least Square	2.41(0.21)	−2.24(0.17)	0.07(0.01)	2.25(0.12)
Tukey's biweight	0.55(0.13)	−0.82(0.12)	0.02(0.01)	1.92(0.04)
Hebert-Leahy	0.34(0.15)	−0.46(0.07)	0.01(0.01)	0.52(0.03)
ANMMS	0.23(0.07)	−0.29(0.07)	0.01(0.01)	0.49(0.03)

Table 2 shows the empirical means and standard deviations upon seven registration trials (Brain atlas images) of four transformations such as translation on both X- axis and Y-axis (T_x, T_y), vertical scaling (K_y) and rotation about the center (θ_x). The proposed method for the registration problem provides smallest standard deviation than the other robust estimators that are applied individually. These results demonstrate the ability of the proposed approach for parameter estimation in image registration problem.

4 Conclusion

A robust parameter estimation technique using acquainted non-convexity multiresolution based minimization scheme *(ANMMS)* for image registration problem has been proposed. The superiority of the proposed approach for robustly estimating the affine parameters has been demonstrated by experimental comparative study with other robust parameter techniques applied on brain atlas images. The performance of the proposed scheme is compared with the affine parameters estimated through various robust estimators that are applied individually. This new proposal considerably improves the power of parameter estimation in all kind of brain atlas images and hence the proposed approach achieves the worthy level of applicability.

References

1. Woods, R.P.: Within-modality registration using intensity-based cost functions. In: Bankman, I. (ed.) Handbook of Medical Imaging: Processing and Analysis, vol. 33, Academic Press, New York (2000)
2. Rossman, M.A., Candocia, F., Adjouadi, M., Jayakar, P., Yaylali, I.: Application of Affine Transformations for the Co-registration of SPECT Images. In: Proceedings of the Fourth IASTED International Conference on Signal and Image Processing (SIP), Kauai, Hawaii, August 12-14, pp. 595–600 (2002)
3. Periaswamy, S., Farid, H.: Elastic Registration in the Presence of Intensity Variations. IEEE Transactions on Medical Imaging 22(7), 865–874 (2003)
4. Black, M.J., Anandan, P.: The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. Computer Vision and Image Understanding 63(1), 75–104 (1996)
5. Negahdaripour, S.: Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(9), 961–979 (1996)
6. Kim, Y.-H., Martinez, A.M., Kak, A.C.: Robust Motion Estimation under Varying Illumination. Image and Vision Computing 23(4), 365–375 (2005)
7. Besl, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
8. Hajnal, J.V., Nadeem, S., Soar, E.J., Oatridge, A., Young, I.R., Bydder, G.M.: A Registration and Interpolation Procedure for Subvoxel Matching of Serially Acquired MR Images. Journal of Computer Assisted Tomography 19(2), 289–296 (1995)
9. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press, Cambridge (1987)

10. Dahyot, R., Charbonnier, P., Heitz, F.: Robust visual recognition of colour images. In: Proceedings of CVPR, vol. 1, pp. 685–690 (2000)
11. Hebert, T., Leahy, R.: A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors. IEEE Transactions on Medical Imaging 8, 194–202 (1990)
12. Farid, H., Simoncelli, E.P.: Optimally rotation-equivariant directional derivative kernels. In: Sommer, G., Daniilidis, K., Pauli, J. (eds.) CAIP 1997. LNCS, vol. 1296, pp. 207–214. Springer, Heidelberg (1997)
13. Johnson, K.A., Becker, J.A.: The whole brain Atlas, `http://www.med.harvard.edu/AANLIB/`
14. Wolberg, G.: Digital image warping. IEEE Computer Society Press, Los Alamitos (1994)
15. Holden, M., Hill, D.L.G., Denton, E.R.E., Jarosz, J.M., Cox, T.C.S., Rohlfing, T., Goodey, J., Hawkes, D.J.: Voxel Similarity Measures for 3D Serial MR Brain Image Registration. IEEE Transactions on Medical Imaging 19(2), 94–102 (2000)
16. Charbonnier, P., Blanc-Fraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Transactions on Image Processing 6(2), 298–311 (1997)
17. Holland, P.W., Welsch, R.E.: Robust regression using iteratively reweighed least-squares. Communications in Statistics: Theory and Methods A6, 813–827 (1977)

An Improved Hybrid Multi-objective Particle Swarm Optimization Algorithm

Zuan Zhou, Guangming Dai, Pan Fang, Fangjie Chen, and Yi Tan

School of Computer, China University of Geosciences,
430074 Wuhan, P.R. China
zhouzuan2k@gmail.com, gmdai@cug.edu.cn

Abstract. Particle Swarm Optimization is a promising evolutionary optimization algorithm. In this paper, an improved hybrid multi-objective particle swarm optimization algorithm (IHMOPSO) is proposed. IHMOPSO uses orthogonal design to initialize population, selects global optimal position from Pareto set. Apply mutation, cross operation and evolutionary selection, and uses two ways to update the position and velocity of particles. Experimental results on many well-known benchmark optimization problems have shown that IHMOPSO is effective and efficient.

Keywords: Particle swarm optimization, multi-objective optimization, orthogonal initialization, Cauchy mutation.

1 Introduction

In many research and application areas, there are a lot of multi-objective problems (MOPs). Commonly two or more objective need to be optimized simultaneously, which always are conflict, so MOPs are very difficult to handle. During the past decades, various population-based approaches, such as genetic algorithms, are used to solve MOPs. Researchers have produced a number of efficient multi-objective evolutionary algorithms, such as NSGA II [1], SPEA II [2].

Particle Swarm Optimization (PSO) is a promising and stochastic optimization techniques developed by James Kennedy and R.C.Eberhart [3]. It is inspired by the intelligence in behaviors of the natural swarms, such as bird colonies trying to reach an unknown destination. Each particle in swarm is one solution of the solution space. Applied to optimization, particles "fly" through the search space by following the previous best positions of their neighbors and their own previous best positions. The position and velocity of each particle are updated according to Eq. (1) and (2):

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pBest_i - X_i(t)) + c_2 r_2 (gBest_i - X_i(t)) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1). \quad (2)$$

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 181–188, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Where X_i and V_i represent the position and the velocity of particle i, $pBest$ and $gBest$ are the previous best position of particle i and the global best position found in the whole swarm so far respectively. $\omega(0 \le \omega < 1)$ is an inertia weight which determines how much the previous velocity is preserved, c_1 and c_2 are acceleration constant which constrain the influence of the $pBest$ and the $gBest$. r_1 and r_2 are two random numbers independently generated within the range of [0,1].

Since PSO and Genetic Algorithm have some similarities, it is a natural extension of applying PSO to MOPs, named multi-objective PSO (MOPSO). Moore and Chapman proposed the first extension of the PSO strategy for solving multi-objective problems in an unpublished manuscript from 1999[4]. After this early attempt, a great interest to extend PSO arose among researchers. There are currently over twenty five different proposals of MOPSOs reported in the specialized literature [5], such as Coello's MOPSO [6], NSPSO [7] and VEPSO [8].

The remainder of this paper is organized as follows. In Section 2, we describe the proposed Improved Hybrid Multi-objective Particle Swarm Optimization (IHMOPSO) algorithm. Section 3 defines the benchmark optimization problems used in the experiments, gives the experimental settings, and discusses the experimental results. Finally, we present our conclusions in Section 4.

2 IHMOPSO Algorithm

Here, we propose an improved hybrid Multi-objective Particle Swarm Optimization algorithm. This algorithm uses orthogonal design to initialize population, apply the method of NSGA Ⅱ to update the archive set, selects global optimal position from Pareto set. Apply mutation, cross operation and evolutionary selection, and uses two ways to update the position and velocity of particles.

2.1 Orthogonal Initialization

Recently, Leung and Zhang [9] observed that some major steps of a genetic algorithm (GA) can be considered to be "experiments." For example, a crossover operator samples the genes from the parents to produce some potential offspring, and this operation can be considered to be a sampling experiment. Numerical results demonstrated that it is very effective in discrete variable optimization, but it is not applicable to continuous variables optimization. In order to solve this problem, Leung, and Wang [10] used a quantum technology to carry out the orthogonal design of continuous variable in single objective optimization problem.

When considering multi-objective optimization problems, if we can get some prior knowledge in the initial stage, we may easily move to the Pareto front. We apply the continuous orthogonal design to generate the initial population.

We denote decision vector $x = (x_1, x_2, \cdots, x_N) \in X$, where x_i is the i_{th} factor in orthogonal experiment. So there are N factors in continuous space. We need to quantize the N factors and divide each factor into Q levels. When Q and N are large, it may not be possible to do all Q^N experiments. Then we design an orthogonal array (OA) from the Q^N combinations and generate the initial population which can scatter uniformly over the feasible solution space. Regularly, the number of the OA's row

is still larger than the population size PS, so we use the ranking mechanism which mention in the NSGA II to select PS solutions from the orthogonal array.

2.2 Cauchy Mutation

Standard PSO algorithm usually has a fast convergence velocity, and can converge to one point of the feasible solution space quickly. But many experiments show that PSO is easy to get trapped, once falling into local optima. Therefore, a mutation operation is needed to jump out of the local optima.

Cauchy distribution is easier to generate a random number which is away from the origin point. This makes the Cauchy mutation is easier to jump out of the local minimum area. The article [11] shows that Cauchy mutation operator has great global search capability. And the article [12] raises a new particle, which is the combination of Cauchy mutation and standard PSO algorithm and has proved effective in the single-objective optimization problem. Therefore this paper uses Cauchy mutation operator to improve the shortcomings of PSO—easy to get trapped in local optima. Besides to update every particle's location and velocity using Eq. (1) and Eq. (2), we use the following two formulas to update:

$$V_i(t+1) = V_i(t)\exp(\delta) \tag{3}$$

$$X_i(t+1) = X_i(t) + V_i(t+1)\delta_i \tag{4}$$

Where δ and δ_i are Cauchy random number.δ is randomly generated in each particles in each dimension, in order to search in each dimension with different scale. δ_i is randomly generated in each particles. We can generate two particles, and select the better one to enter the offspring population by calculating its pros and cons, using Eq. (1), Eq. (2), Eq. (3) and Eq. (4).

2.3 Cross and Selection

After years of research and improvement, Genetic Algorithm has developed maturely. We can learn some effective idea from it to improve PSO, for instance, crossover and selection are two kinds of typical genetic algorithm method.

Crossover operation, which makes children individual inherit the merits of the parents individual, may enhance the search capabilities of algorithms theoretically. In this paper, each particle X_i randomly selects a non-inferior particle X'_i as crossover target. And then, do the crossover operation with Eq. (5) and Eq. (6):

$$V_i = \alpha(X'_i - X_i) \tag{5}$$

$$X_i = (1-\beta)X_i + \beta X'_i \tag{6}$$

Where α and β are random number within the range of (0, 1).

Natural selection reflects Darwin's biological evolution rule—survival of the fittest, which is the results from billion years of nature evolution. We import this

strategy to PSO, making all the particles survival or elimination according to the rule, and ensure the particles surviving are the most adaptive to the environment.

We hybridize current population, offspring population and crossover population all of whose population size are M, sort hybrid population using the fast non-dominate sorting mechanism in NSGA-Ⅱ, select first M individuals as the individual in the new population, which will largely avoid getting trapped in local optima.

2.4 Algorithm Procedure

The detailed steps of IHMOPSO are listed as follows.

Step1. Initialize the current population (current_pop), the offspring population (next_pop), the cross population (cross_pop) and the mixture population (mix_pop).Their size are M、M、M、3M：

1) Orthogonal initialize the position of each particle in current_pop, and set the velocity threshold value as 0. Then Calculate the fitness of each particle in current_pop..

2) Set the position and velocity of each particle in next_pop, cross_pop and mixpop.

Step 2: Generate the offspring population (next_pop).

1) Randomly select the globe best of each particle from the non-dominate set of current_pop.

2) For each particle in current_pop, do:

a) Generate a new particle p_1 with formula (**1**) and (**2**).

b) Generate another new particle p_2 with formula (3) and (4).

c) Compare p_1 and p_2 , select the better one to enter next_pop.

Step 3: For each particle p in current_pop, randomly select a particle p', crossover particle p with p' by formula (5) and (6), and let the new particle enter cross_pop.

Step 4: Merge the current population (current_pop), the offspring population (next_pop) and the cross population (cross_pop) into the mixture population (mix_pop).Sort mix_pop by the fast non-dominate sorting mechanism in NSGA-Ⅱ and select the first M particles enter current_pop.

Step 5: If the stop criterion is not satisfied, then goto Step 2, else stop and output the non-dominate particles as the result of the algorithm.

3 Experiments and Results

Five test problems (ZDT1-ZDT4, ZDT6)[13] are used in this study to investigate the IHMOPSO algorithm. All of them are described as follows.

The algorithms used for comparison were NSGA-Ⅱ, SPEA, PAES, OMEA and EM-MOPSO. Each algorithm was tested with test problems shown in Table 1.The parameters and results of NSGA-Ⅱ, SPEA and PAES come from article [1]. The

Table 1. Test Problems used in this study. All objective functions are to be minimized.

Problem	n	Variable bounds	Objective functions
ZDT1	30	[0,1]	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left[1-\sqrt{x_1 / g(\mathbf{x})}\right]$ $g(\mathbf{x}) = 1+9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT2	30	[0,1]	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left[1-(x_1 / g(\mathbf{x}))^2\right]$ $g(\mathbf{x}) = 1+9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT3	30	[0,1]	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left[1-\sqrt{x_1 / g(\mathbf{x})}-\frac{x_1}{g(\mathbf{x})}\sin(10\pi x_1)\right]$ $g(\mathbf{x}) = 1+9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT4	10	$x_1 \in [0,1]$ $x_i \in [-5,5]$ $i = 2,...,n$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})\left[1-\sqrt{x_1 / g(\mathbf{x})}\right]$ $g(\mathbf{x}) = 1+10(n-1)+\sum_{i=2}^{n}\left[x_i^2-10\cos(4\pi x_i)\right]$
ZDT6	10	[0,1]	$f_1(\mathbf{x}) = 1-\exp(-4x_1)\sin^6(4\pi x_1)$ $f_2(\mathbf{X}) = g(\mathbf{X})\left[1-(f_1(\mathbf{x}) / g(\mathbf{x}))^2\right]$ $g(\mathbf{x}) = 1+9\left[(\sum_{i=2}^{n} x_i)/(n-1)\right]^{0.25}$

parameter and result of OMEA comes from article [14] and EM-MOPSO comes from article [15]. Especially the maximum generation is set as 100, 500 for OMEA and EM-MOPSO respectively, and set as 250 for others.

In the experiment, the maximum population is 100, acceleration constants of c_1 and c_2 are both set to be 1.496180. The inertia weight ω is a self-adaptive parameter, which is calculated as follows [16]:

ω = population size / (population size + particle number dominated by the particle).

All of the algorithms for each test problem were run over 100 times. We use the Convergence Metric and Diversity Metric which were proposed by Deb [1] to evaluate the results. The statistical results of each algorithm for these problems are shown in Table 2 and Table 3.

Table 2. Mean (shaded rows) and variance (unshaded rows) of the convergence metric

Problem	NSGA-Ⅱ	SPEA	PAES	OMEA	EM-MOPSO	IHMOPSO
ZDT1	0.000894	0.001799	0.082085	0.001220	0.005130	0.001049
	0.000000	0.000001	0.008679	0.000144	0.000120	0.000368
ZDT2	0.000824	0.001339	0.126276	0.000814	0.004590	0.000808
	0.000000	0.000000	0.036877	0.000077	0.000390	0.000049
ZDT3	0.043411	0.047517	0.023872	0.001259	0.007200	0.001278
	0.000042	0.000047	0.000010	0.000079	0.000380	0.000103
ZDT4	3.227636	7.340299	0.854816	0.000968		0.001167
	7.307630	6.572516	0.527238	0.000103		0.000147
ZDT6	7.806798	0.221138	0.085469	0.000891	0.006320	0.000579
	0.001667	0.000449	0.006664	0.000309	0.001320	0.000033

Table 3. Mean (shaded rows) and variance (unshaded rows) of the diversity metric

Problem	NSGA-Ⅱ	SPEA	PAES	OMEA	EM-MOPSO	IHMOPSO
ZDT1	0.463292	0.784525	1.229794	0.352306	0.245020	0.415356
	0.041622	0.004440	0.004839	0.0272603	0.019230	0.142073
ZDT2	0.435112	0.755148	1.165942	0.353621	0.289770	0.472190
	0.024607	0.004521	0.007682	0.0239713	0.021810	0.056309
ZDT3	0.575606	0.672938	0.789920	0.541039	0.760130	0.545236
	0.005078	0.003587	0.001653	0.019943	0.041740	0.020896
ZDT4	0.479475	0.798463	0.870458	0.376844		0.452438
	0.009841	0.014616	0.101399	0.0177152		0.039772
ZDT6	0.644477	0.849389	1.153052	0.550322	0.533920	0.427326
	0.035042	0.002713	0.003916	0.041836	0.120850	0.026652

Figure 1 shows the result of IHMOPSO for ZDT1, ZDT3, ZDT4 and ZDT6.

By viewing the result of Table 2 and Table 3, we know that NSGA-Ⅱ, SPEA and PAES can not solve ZDT4 problem and ZDT6 problem effectively. It's is because the trait of these two problems: ZDT4 is non-convex and have 21^9 local Pareto fronts; ZDT6 is non-convex, non-uniformly and spaced. When discussing the convergence, we can easily see that IHMOPSO show better performance than NSGA-Ⅱ, SPEA, PAES and EM-MOPSO on all these problems except for the simplest problem-ZDT1. Especially, IHMOPSO is better than another MOPSO algorithm –EM-MOPSO. It shows the mechanism of orthogonal initialization and Cauchy mutation, applying in MOPSO, speeds up the search. When discussing the diversity, OMEA shows a better performance in most problems than IHMOPSO, but it is high time cost. Usually,

the time cost of OMEA is almost 100 times than others when calculating for the same maximized generation. However, for the hardest problem-ZDT6, OMEA is poorer than IHMOPSO in both convergence metric and diversity metric. EM-MOPSO shows a better performance for ZDT1 and ZDT2, but it is poorer than IHMOPSO for ZDT3 and ZDT6.

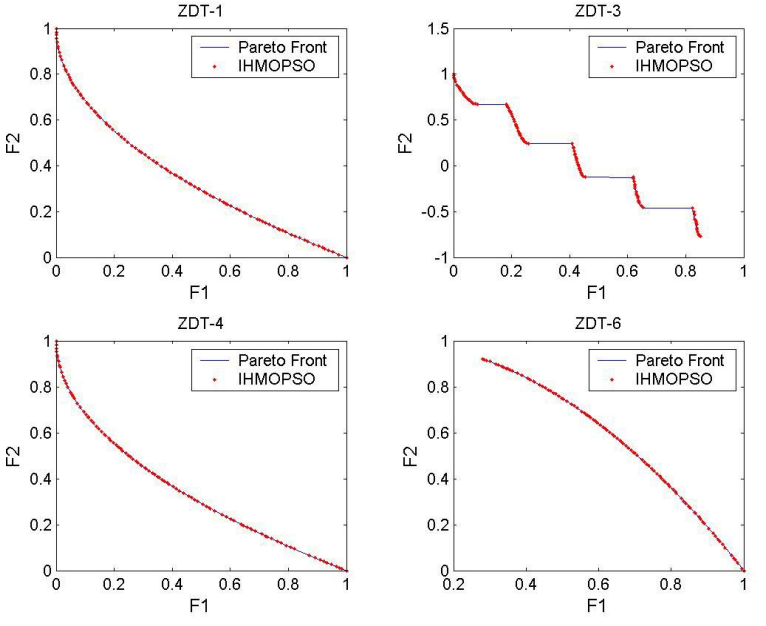


Fig. 1. The results of IHMOPSO for ZDT1, ZDT3, ZDT4 and ZDT6

From what has discussed above, we can come to the conclusion that when testing ZDT problem set, IHMOPSO algorithm is not only superior to the classical NSGA- II and SPEA , but also better than EM-MOPSO which is the newer MOPSO algorithm.

4 Conclusions

This paper proposes an improved hybrid multi-objective particle swarm optimization algorithm (IHMOPSO). IHMOPSO uses orthogonal design to initialize population, selects global optimal position from Pareto set. Apply mutation, cross operation and evolutionary selection, and uses two ways to update the position and velocity of particles in order to increase the probability of jumping out of the local optimum. The algorithm was tested in five famous ZDT problems. From the result, we can see IHMOPSO shows a competitive performance in convergence and divergence generally.

References

1. Deb, K., Pratap, A., Agarwal, S., et al.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
2. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory(TIK), Swiss Federal Institute of Technology(ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)
3. Kennedy, J., Eberhar, R.: Particle Swarm Optimization. In: Proceedings of the Fourth IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
4. Moore, J., Chapman, R.: Application of particle swarm to multi-objective optimization. Department of Computer Science and Software Engineering, Auburn University (1999)
5. Reyes-Sierra, M., Coello Coello, C.A.: Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. International Journal of Computational Intelligence Research 2(3), 287–308 (2006)
6. Coello, C., Lechunga, M.: MOPSO: A proposal for Multiple Objective Particle Swarm Optimization. In: Proceedings of 2002 Congress on Evolutionary Computation, pp. 1051–1056. IEEE Press, Los Alamitos (2002)
7. Li, X.: A nondominated sorting particle swarm optimizer for multi-objective optimization. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 37–48. Springer, Heidelberg (2003)
8. Parsopoulos, K., Vrahatis, M.: Particle Swarm Optimization Methods in Multiobjective Problems. In: Proceedings of 2002 ACM Symp. Applied Computing (SAC 2002), Madrid, Spain, pp. 603–607 (2002)
9. Leung, Y.W., Zhang, Q.: Evolutionary algorithms experimental design methods: A hybrid approach for hard optimization and search problems, Res. Grant Proposal, Hong Kong Baptist Univ. (1997)
10. Leung, Y.-W., Wang, Y.: An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation 5(1) (2001)
11. Yao, X., Liu, Y.: Fast evolutionary programming. In: Proc. Of the Fifth Annual Conference on Evolutionary Programming (EP 1996), pp. 451–460. MIT Press, San Diego (1996)
12. Li, C., Liu, Y., Zhou, A., et al.: A Fast Particle Swarm Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy. In: Kang, L., Liu, Y., Zeng, S. (eds.) ISICA 2007. LNCS, vol. 4683, pp. 344–352. Springer, Heidelberg (2007)
13. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications
14. Zheng, W.: Research and Application of OMEA in the Optimal Design of Constellation. Master thesis. School of Computer, China University of Geosciences, Wuhan, Hubei, China
15. Reddy, M.J., Kumar, D.N.: An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. Engineering Optimization 39(1), 49–68 (2007)
16. Zhang, Q., Xue, S.: An Improved Multi-Objective Particle Swarm Optimization Algorithm. In: Kang, L., Liu, Y., Zeng, S. (eds.) ISICA 2007. LNCS, vol. 4683, pp. 372–381. Springer, Heidelberg (2007)

Application of Ant Colony Optimization Algorithm to Multi-Join Query Optimization

Nana Li, Yujuan Liu, Yongfeng Dong, and Junhua Gu

College of Computer Science and Software, Hebei University of Technology, Tianjin 300401, China
{Na-Na Li,Yu-Juan Liu,Yong-Feng Dong,Jun-Hua Gu, cepits}@163.com

Abstract. Multi-join query optimization (MJQO) is an important technique for designing and implementing database manage system. It is a crucial factor that affects the capability of database. This paper proposes a new algorithm to solve the problem of MJQO based on ant colony optimization (ACO). In this paper, details of the algorithm used to solve MJQO problem have been interpreted, including how to define heuristic information, how to implement local pheromone update and global pheromone update and how to design state transition rule. After repeated iteration, a reasonable solution is obtained. Compared with genetic algorithm, the simulation result indicates that ACO is more effective and efficient.

Keywords: Multi-join query optimization, ant colony optimization algorithm (ACO), pheromone, heuristic information.

1 Introduction

Multi-join query optimization (MJQO) is a difficulty in relational database manage system (RDBMS) which has been solved faultily. In traditional applications of RDBMS, the number of join N involved by a single query is relatively small, Usually, N<10. With the expansion of the database application areas, the traditional query optimization technology cannot support some of the latest database applications. Such as, applications of decision support system (DSS), OLAP and data mining (DM), which may produce a query including more than 100 joins. In this condition, the shortfall of the traditional query optimization technology is exposed gradually. Therefore, it is necessary to explore new technology to solve MJQO problem.

MJQO is an NP hard problem. With the increase of join number, the number of query execution plan (QEP) corresponding to a query grows exponentially, which lead to computational complexity of MJQO problem is very large. Recently, solving the problem with heuristic algorithm becomes a hotspot. Such as, Greedy Algorithm[1], KBZ Algorithm[2],GA[3], AB[4] etc. ACO as a highly effective optimization algorithm has been successfully applied to some classic compounding optimization problems, such as, the traveling salesman problem (TSP) [5], quadratic assignment problem (QAP)[6], job - shop scheduling problem(JSP)[7], vehicle routing problem(VRP) [8], etc. In this paper, ACO was adopted to solve the problem MJQO.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 189–197, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Description for Multi-Join Query Optimization Problem

The process of RDBMS managing user query is as follows: After receiving query submitted by users, query parser checks syntax, verifies relations, translates the query into its internal form. It is usually translated into relational algebra expression, which can be denoted as query syntax tree. A relational algebra expression may have many equivalent expressions, so it also corresponds to many equivalent query syntax tree. Then, query optimizer selects appropriate physical method to implement each relational algebra operation and finally generate query execution plan (QEP). The QEP consists of the order in which the operations in a query are to be processed, and the physical method to be used to process each operation. Amongst all equivalent QEP, query optimizer chooses the one with lowest cost output to the query-execution engine, Then, the query-execution engine takes the QEP, executes that plan, and returns the answers to user. The process is depicted in Fig 1. This paper is to study how to make query optimizer select a QEP with lower cost in shorter time.

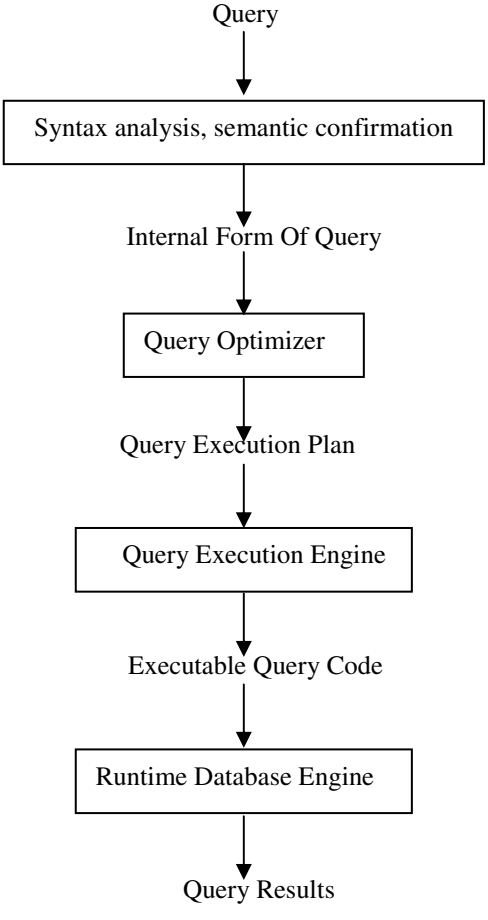


Fig. 1. Process of query execution

After being implemented optimizing operation of pushing down the select operation and project operation, one query that include project, select and join operations will transform to relational algebra expression constituted by N join operations which can be denoted as join tree. An example multiple query Q include A, B, C, D, E five relations, which can be denoted as three kinds of join tree shown in Fig 2: (a) left-deep tree (b) bushy tree (c) right-deep tree. The leaf nodes are relations constituting query Q and the internal nodes express join operation and intermediate results. Executive order is bottom-up execution. The different order of N join and the different physical methods selected to implement join operation lead to the cost of join trees have great differences. Assume that each join operations are implemented by the same physical method; Multi-join query optimization problem is simplified as setting a good join order, making the join tree has the lowest cost. Hence, tree in the left linear space can take full advantage of the index, and often contain the best strategy or the strategy whose cost is similar to the best strategy at least, therefore, consider left linear space as a search space.

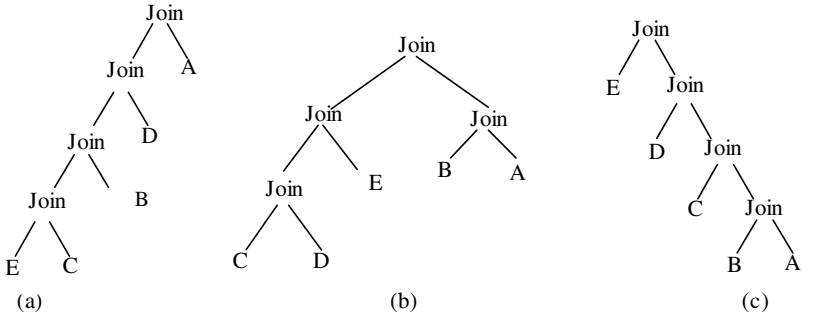


Fig. 2. Three kinds of join tree

In order to reduce the search space further more, avoiding the emergence of Cartesian product often is considered as constraint of the issue. An example multiple join query Q include A, B, C, D, E five relations. The attributes associating between five relations which are founded from statistical information of the database catalog, could be denoted as a query graph G=(V, E), shown as Figure 3. Nodes in query graph are relations and an edge connecting two relations, indicates attributes associating between two relations.

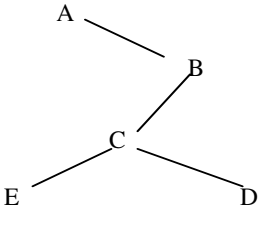


Fig. 3. Query graph

Q1 and Q2 are two join trees in the left linear space of query Q, depicted in Figure 4. Taking into account the "avoiding Cartesian product " restrictive conditions, Q1 do not accord with the restrictive conditions and Q1 is invalid join tree; Q2 do accord with the restrictive conditions and Q2 is valid join tree.

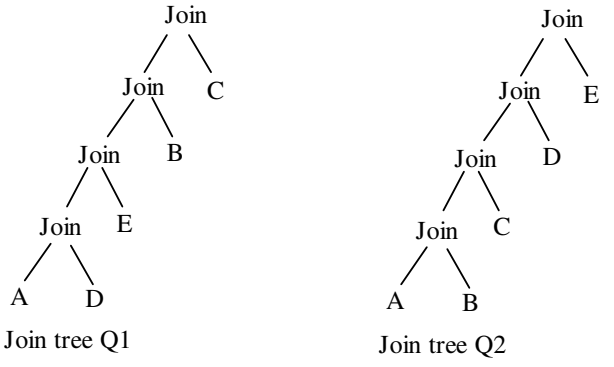


Fig. 4. Two join trees

Each relation in query graph corresponds to a set of parameters given by:

$n(r)$: Tuples number of relation r

$V(C,r)$: Number of distinct values for attribute C in relation r

In this paper, a simple model of the estimated cost is used, which based on two assumptions: Firstly, attribute values in symmetrical distribution. Secondly, the sum of the tuples number about intermediate results decides the cost of QEP. For example, t = r join s, C is public attribute over r, s. Then, $n(t)$ and $V(A,t)$ are defined by the following formulas:

$$n(t) = \frac{n(r) \times n(s)}{\prod_{C_i \in C} \max(V(C_i, r), V(C_i, s))} \tag{1}$$

$$V(A,t) = \begin{cases} V(A,r) & A \in r - s \\ V(A,s) & A \in s - r \\ \min(V(A,r), V(A,s)) & A \in r, A \in s \end{cases} \tag{2}$$

Assume that there are N relations in a join tree; the cost of QEP is the sum of the tuples number of internal nodes t_i in join tree. $n(t_i)$ is number of tuples about

intermediate result t_i. For a query Q, Z is collection of all the possible QEP corresponding to Q. Each member z in collection Z has query execution cost--- $Cost(z)$, then, z_0 meeting $Cost(z_0) \approx \min_{z \in Z} Cost(z)$ should be found.

3 Ant Colony Optimization Algorithm for MJQO Problem

The ACO algorithm, introduced by Marco Dorigo in 1992 in his PhD thesis[9], is a probabilistic technique for solving computational optimization problems which can be reduced to finding good paths through graphs. They are inspired by the behavior of ants in finding paths from the colony to food.

As said in the previous section, application of an ACO algorithm to MJQO problem requires definition of a constructive algorithm. Accordingly, a constructive algorithm called ACO-MJQO is proposed in which a set of artificial ants builds feasible solutions with some join order. Because of positive feedback mechanism, the ant could collaborate with each other and could find the better solution in shorter time. ACS-MJQO works as follows:

```
Input: Collection of all relations corresponding to
       query Q called L.
Output: Join tree M with special join order making
       query execution cost lowest
Begin
Each ant has been placed to the same relation R;
 /*Number of tuples about R is smallest*/
For i=1 to m do {/*m is iteration number * /
  For j=1 to n do {/* n is ants number* /
     M=R;
     While (L is not null) {
        if(relation C and current relation node are
          linked with edge in query graph){
           According to state transition rule choose
           next relation node C;
           M =M∪C;
           Update query graph, delete edge E(Ri,C),
           Ri is current node;
           Ri and C are merged into as current rela-
           tion node;
           L = L-C;
           Local update pheromone;
          } //end if
           }//end while
     n=n+1;
   } //end For j=1 to n
  Record the cost of optimal solution Q(m) in this
  iteration;
  Global update pheromone;
  m=m+1;
}
Record global optimal solution M;
End
```

3.1 State Transition Rule

In order to construct complete solutions, each ant iteratively starts from a relation node and adds new relation node until all relation nodes have been put into the join tree. When in node i, an ant applies transition rule, that is, it probabilistically chooses the next node j from the set US of feasible nodes. Transition rule is as follows:

$$v = \begin{cases} \arg\max_{j \in US}[(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}], q \le q_0 \\ V, \qquad\qquad q > q_0 \end{cases} \tag{3}$$

The set US contains all the nodes which have not been visited and if add them in solution, could not generate Cartesian product. As interpreted in the formula (3), The ant in node i chooses the next node j to visit on the basis of two factors: the heuristic desirability η_{ij} here defined as $d(j)/n(j)$. $d(j)$ is degree of node j in query graph, $n(j)$ is tuples number about node j. The pheromone trail τ_{ij} that contains a measure of how well it has been in the past to include arc (i, j) into a solution (it is the memory of the colony). The parameters α and β control the relative importance of the pheromone versus the heuristic information; q is random variable uniformly distributed over [0,1]. The next node to visit is chosen with probability q0 as the node j, $j \in US$, for which the product $(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}$ is highest (deterministic rule), while with probability 1-q0 the node j is chosen with a probability given by:

$$P_{ij} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{j \in US}(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}, j \in US \\ 0, \qquad\qquad otherwise \end{cases} \tag{4}$$

3.2 Pheromone Updating Rule

In ACO-MJQO, update pheromone in two stages: local pheromone update and global pheromone update.

All of ants perform local pheromone update after each construction step. Each ant, when moving from node i to node j, applies a pheromone updating rule given by:

$$\tau_{ij}^{new} = \rho * \tau_{ij}^{old} + (1-\rho) * \tau_0 \tag{5}$$

Where τ_0 is the pheromone decay coefficient, and is the initial value of the pheromone, which can be defined as $\tau_0 = 1/N * C_{\min}$, where N is total number of relations related to query Q, $C_{\min}$ is cost of optimal solution solved by greedy algorithm.

The main goal of the local update is to diversify the search performed by subsequent ants during an iteration: by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and, hence, to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration.

The global pheromone update is applied at the end of each iteration by only one ant, which can be either the iteration-best or the best-so-far. The update formula is as follows:

$$\tau_{ij}^{new} = \rho * \tau_{ij}^{old} + (1-\rho) * \Delta\tau_{ij} \tag{6}$$

The rationale is that in this way a "preferred route" is memorized in the pheromone trail matrix, and future ants will use this information to generate new solutions in a neighborhood of this preferred route. $\Delta\tau_{ij}$ is defined as $\Delta\tau_{ij} = 1/Q(M)$, where Q(M) is the cost of optimal solution in one iteration.

4 Experimental Results

In order to illustrate the effect of ACS-MJQO in solving this problem, experiments have been implemented on computer with Pentium4 2.93G+512RAM+Windows2000. A database including 50 relations that have attributes association with each other has been used as test data. Genetic algorithm and ACO are used to solve this problem respectively.

The values of parameters in ACS-MJQO are as follows: α=1, β=3, q0=0.2, $\rho = 0.9$, ants number m=10.

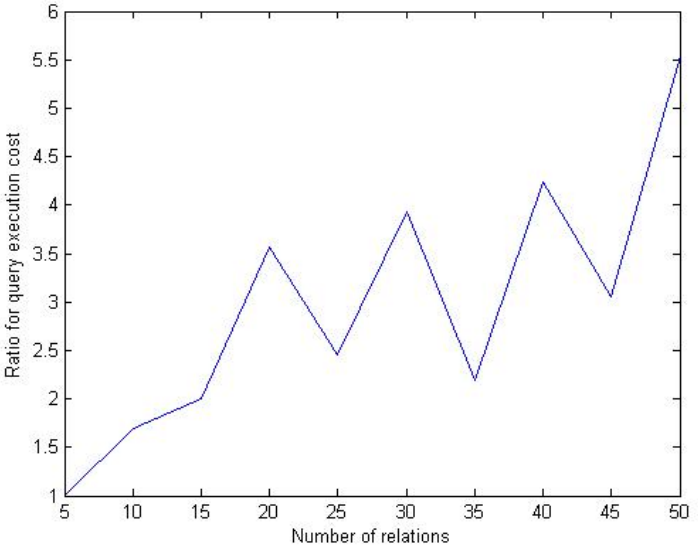


Fig. 5. Ratio of query execution cost

As is indicated in Fig 5, number of relations corresponding to query Q are taken as X-axis and values defined as query execution cost of GA/ query execution cost of ACO, are taken as Y-axis, Fig 5 indicates the cost of optimal solution generated by ACO is lower than that generated by GA.

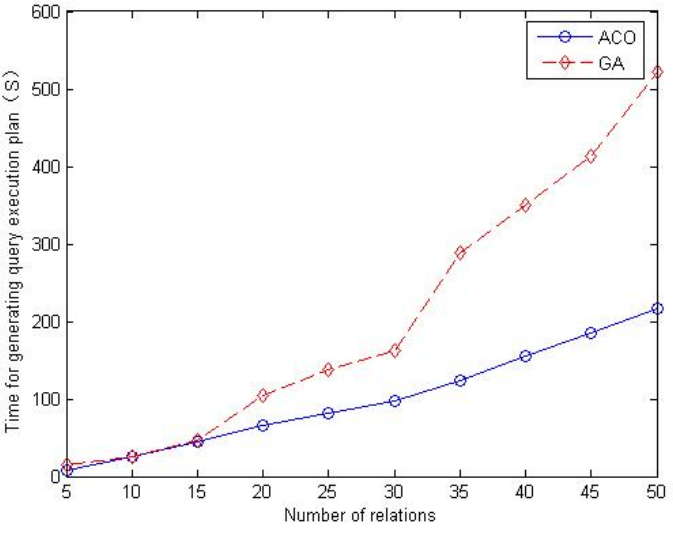


Fig. 6. Comparisions of execution time

As is shown in Fig 6, number of relations corresponding to query Q are taken as X-axis and time for generating optimal solution—query execution plan is taken as Y-axis. The simulation results show that ACO finds optimum solutions more effectively both in time and quantity than GA The figure indicates genetic algorithm spends more time than ACO algorithms on finding optimal solution especially with the incensement of relation number.

5 Conclusion

MJQO problem is hotspot in database research field. A good optimization algorithm not only can improve the efficiency of queries but also reduce query execution costs. In this paper ACO algorithm was proposed to solve the problem of MJQO. In ACO-MJQO, local pheromone update and global pheromone update are applied, and at last obtain a rational solution.

The simulation results show that ACO finds optimum solutions more effectively both in time and quantity than GA especially with the incensement of relation number.

References

1. Shekita, E., Young, H., Tan, K.L.: Multi-join optimization for symmetric multiprocessors. In: Proc. Of the Conf. on Very Large Data Bases (VLDB), Dublin, Ireland, pp. 479–492 (1993)
2. Krishnamurthy, W.R., Boral, H., Zaniolo, C.: Optimization of nonrecursive queries. In: Proc. Of the Conf. On Very Large Data Base (sVLDB), Kyoto, Japan, pp. 128–137 (1986)

3. Cao, Y., Fang, Q.: Parallel Query Optimization Techniques for Multi-Join Expressions Based on Genetic Algorithmspline. Journal of Software 13(2), 250–256 (2002)
4. Swami, A., Iyer, B.: A polynomial time algorithm for optimizing join queries. In: Proc. IEEE Conf. on Data Engineering, Vienna, Austria, pp. 345–354 (1993)
5. Dorigom, M., Gambardella, L.M.: Ant colonies for the traveling salesman problem. BioSystems 43(2), 73–81 (1997)
6. Maniezzo, V., Dorigo, M., Colorni, A.: The ant system applied to the quadratic assignment-problem, IRIDIA/94-28. Universite de Bruxelles, Belgium (1994)
7. Colorni, A., Dorigo, M., Maniezzo, V., et al.: Ant system for job-shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science 4(1), 39–53 (1994)
8. Bullnheimer, B., Hartl, R.F., Strauss, C.: Applying the ant system to the vehicle routing problem. In: Osman, I.H., Vo, S., Martello, S., Roucairol, C. (eds.) Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, pp. 109–120. Kluwer Academics, Dordrecht (1998)
9. Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics–Part B 26(1), 29–41 (1996)

A Hybrid Algorithm for Vehicle Routing Problem with Time Windows

Dengying Jiang, Wenxia Jiang, and Zhangcan Huang

School of Science, Wuhan University of Technology, Wuhan 430070, China
huangzc@whut.edu.cn

Abstract. The VRPTW is a well-known and complex combinatorial problem which has considerable attention in recent years. Combinatorial optimization problems of this kind are NP-hard and are best solved to near optimum by heuristics. Here, we propose a two-stage optimization strategy for VRPTW. Firstly, for constructing a good initial solution, we use the stochastic PFIH, guaranteeing the diversity of the initial solution. And then a hybrid system based on the combination of SA and LNS is proposed to optimize the initial solution. Secondly, tune time windows for the customers using a regression iterative strategy which is put forward to and figure out the optimum time for each vehicle departing. It can make the total waiting time zero. The test of this work is executed over the type C101 in Solomon's VRPTW instances. It is proved by the experiment that our algorithm can solve VRPTW efficiently and quickly.

Keywords: stochastic PFIH, iterative strategy.

1 Introduction

Vehicle Routing Problem(VRP)is a generic name referred to a class of combinatorial problem in which customers are to be served by a number of vehicles. It is an important element of many logistic systems. The Vehicle Routing Problem with time windows (VRPTW) is a constrained version of the VRP wherein each customer must be served within a specified time window. Instances of VRPTW occur frequently in many industries, e.g., fast-food deliveries, product deliveries, postal deliveries, school bus routing. Even slight improvements in solutions may result in large cost saving. Thus, VRPTW problem is of growing interest in management science, logistics management and computer science.

However, the VRPTW problem is NP-hard. Therefore, present studies of this problem attempt to apply heuristic techniques to obtain suboptimal solutions and solve the problem of getting trapped in local optimal. In these heuristics, a hybrid approach is commonly used. Oliveriva H.C.B. [3] proposed a different approach based on the combination of simulated annealing and a hill climbing strategy with random restart (multi-start). Alvarenga.G.B.[4] proposed a robust heuristic approach using an efficient genetic algorithm and a set partitioning formulation. Andrew. L [5] proposed a two-stage algorithm for m-VRPTW ,firstly maximized the number of customers served with an Ejection Pool to hold temporarily unserved customers, then minimized

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 198–205, 2008.
© Springer-Verlag Berlin Heidelberg 2008

the total travel distance using a multi-start iterated hill-climbing algorithm with classical and new operators including Generalized Ejection Chains.

In our paper, we firstly construct an initial solution by the stochastic PFIH, and then optimize the initial solution using the hybrid algorithm that combines Simulated Annealing and Large Neighbor Search strategy. Finally, a regression iterative strategy is put forward to adjust time windows for the customers and figure out the optimum time for each vehicle departing, it can make the total waiting time zero.

2 Procedure for Paper Submission

In the VRPTW, every customer i, $i \in C = \{1,2,...,n\}$ has a given demand q_i. Our aim is to find the route for every vehicle in such a way that: (1) each customer in C is visited exactly once within its service time; (2) all routes start at the node 0 and end at node $n+1$, $N = \{0,1,...,n+1\}$; (3) the total of customers' demands for each route cannot exceed the vehicle's capacity q, all the vehicles are of the same type and have the same capacity; (4) each customer i has a service time S_i and a service-time window $[e_i, l_i]$, that is, the vehicle services the customer i must arrive before l_i. If it arrives before e_i, it should wait for w_i. Then, basically, the object of the VRPTW is to find a set of customers attended for each vehicle of the fleet in order to minimize the total travel distance or minimize the number of the vehicles used.

Our formulation is based upon the model defined by Solomon [2].

$$\min \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} c_{ij} x_{ijk} \tag{1}$$

$$\sum_{j=1}^{N} x_{ijk} = \sum_{j=1}^{N} x_{jik} \le 1 \quad i = 0, k \in \{1,2\cdots,K\} \tag{2}$$

$$\sum_{k=1}^{K} \sum_{j=0}^{N} x_{ijk} = 1 \quad i, j \in \{0,1\cdots,n\} \tag{3}$$

$$\sum_{i=0}^{N} q_i \sum_{j=0}^{N} x_{ijk} \le q \quad k \in \{1,2\cdots,K\} \tag{4}$$

$$\sum_{k=0}^{K} \sum_{i=0}^{N} x_{ijk} (b_i + s_i + t_{ij} + w_i) = b_j, j \in \{1,2\cdots,n\} \tag{5}$$

$$e_i \le (a_i + w_i) \le l_i \quad i \in N \tag{6}$$

where S_i: the service time of customer i; q_i: the demand of customer i; t_{ij}: the direct travel time between i and j; c_{ij}: cost of traveling from customer i to customer j; a_i: arrival time to customer i; b_i: the time to begin servicing customer i.

If vehicle k travels directly from i to j, $x_{ijk} = 1$; otherwise, $x_{ijk} = 0$, $i \neq j$, $i, j \in C$. The object is to minimize the total vehicle travel cost (1) subject to vehicle capacity, travel time and arrival time feasibility constraints. (2) make sure that each vehicle starts from node 0 and ends at node $n+1$. In addition, each customer can be served by one and only one vehicle. (3) At the same time, the demands of all the customers

serviced by vehicle k cannot exceed the maximum capacity of the vehicle k. (4) Expressions (5)-(6) define the time windows.

3 A Hybrid System for the VRPTW

This work aims at constructing a hybrid system based on the combination of simulated annealing and large neighbor search strategy. For constructing a superior initial solution for SA algorithm, this work uses stochastic PFIH. Readers may refer to Marius Solomon [2] for details of the PFIH algorithm.

The Initial Solutions

To construct the initial routes, the stochastic PFIH also proposed in [1] is utilized. It can produce quickly diversified solutions .The original PFIH [2] is deterministic, but differently, in the stochastic PFIH a random choice is used to define the first customer for each new route. That is necessary to produce diversiform initial solution.

The Procedure to Produce New Solution

The procedure mechanism of the new solution is betterment of large neighbor search (LNS), which is proposed by Pawl Shaw (1998). It starts from the initial solution, and find the optimum solution based upon the processes of continual re-move and re-insertion. Though LNS possesses the competitive search technique in the problems which have complex restraint conditions, but we need a superior initial solution, so we construct the initial solution by Stochastic PFIH. In view of the peculiarity for VRPTW, the processes of re-move and re-insertion are described as follows.

Here, set A is the present solution, c is the customer which will be removed from A, set R is made of the customers removed from the A, p is the number of the customers removed, we use set B to denote the part-solution that when p customers have been removed from A.

The first element of R is the customer z which removed from A at random, the second one chosen from the rest customers in A is who would be maximal related to customer z, the others would be chosen by the relatedness with set R. Every time the customer removed is maximally related to set R. The above-mentioned procedure will be repeated p-2 times until all the required customers are chosen, we use the simple relatedness function $r(i,j)$ to denote the relatedness between any two customers i and j, $R(i,R)$ denote the relatedness between any customer i and R:

$$r(i,j) = 1/(t_{ij}' + v_{ij}) \tag{7}$$

$$R(i,R) = \sum_{j \in R} r(i,j) \tag{8}$$

where $t_{ij}' = t_{ij} / \max(t_{ij})$, v_{ij}=1 if i and j are served by the same vehicle and 0 otherwise; t_{ij} is the time of getting from i to j (travel distance in this paper).

For the sake of finding a new superior solution, we use re-insertion process ,that is, re-insert the element in R to B. But there will be many re-insertion positions for the

customers in R when we insert them to B, so we design the following algorithm in order to guarantee the feasibility of the re-insertion process.

Firstly, fix the best insertion position for every customer in R: the best feasible insertion position for a un-insertion customer is the one that minimize the increment of objective function value, and record the objective function value. After that, sequence all the objective function values just recorded, and choose the customer c which has the minimal increment value. Then re-insert it to B, and remove it from R. At the same time, search all the solutions that can re-insert other customers in R to B and repeat above procedure until R is empty, that is, all the customers removed from B have been inserted B again. Then we compare all the solutions we have found with B. If B is superior, choose B as the optimum solution present.

The Integrated Algorithm for VRPTW

The hybrid algorithm in this paper consists of two stages: firstly, the stochastic PFIH is utilized to produce a finer initial solution; secondly, we extend the search space to avoid trapping in a local optimum, and combine LNS and SA to optimize the tour for finding the optimum solution. The whole procedure is described as follows:

Step 1: Produce an initial solution S_0 by stochastic PFIH and calculate objective function value $f(S_0)$. Fix initial temperature $T=T_0$. For each T, the iterative time is l whose maximum value is L, m is the times that don't accept continuously new solution by now, and the upper bound of m is M.

Step 2: If $T > T'$ (T' is the termination temperature), then turn to step 3, otherwise to step 6.

Step 3: If $l<L$ and $m<M$, then turn to step 4; if $m>M$, turn to step 6.

Step 4: Produce a new solution S', calculate $\Delta f = f(S') - f(S_0)$, if $\Delta f < 0$, then the new solution S' is automatically accepted to replace S_0, then turn to step3; otherwise, accepting the new solution S' will depend on the probability established by the Metropolis Criteria, which is given by $e^{\Delta f/T}$, if accept, then $m=0$ and turn to step 3, otherwise $m=m+1$, turn to step 3.

Step 5: T decreases successive depend on formula $T_{k+1}=\alpha T_k$, turn to step 2.

Step 6: Export the optimum solution.

4 The Regression Strategy for Time Windows of Customers

Because the target of the feasible route is not only to minimize the total traveled distance, but also to minimize the total wait time, even to zero. Thus, in our paper, minimizing the total traveled distance is taken as the first target, and we also consider the waiting cost in the traffic, and put forward a strategy to adjust the time windows for the customers and figure out the best time for each vehicle departing, so it can make the total waiting time zero. We define it regression iterative strategy as follows:

Let "$0 \to v_1 \to \cdots v_n \to 0$" is the feasible route for vehicle K, $[e_{v_i}, l_{v_i}]$ is the time window for customer v_i. After the time window is adjusted, it becomes $[e'_{v_i}, l'_{v_i}]$,

$v_i \in \{1,2,...,n\}$. $[e_0, l_0]$ is the time window of vehicle *K*. The step for adjusting the time window for each customer in the whole route is:

Step1: calculate $e'_{v_n} = \max\{e_{v_n}, e_0 - t_{0,v_n} - s_{v_n}\}$, $l'_{v_n} = \min\{l_{v_n}, l_0 - t_{0,v_n} - s_{v_n}\}$.

Step2: updating the time window of v_n as $[e'_{v_n}, l'_{v_n}]$, repeat step 1 , calculate the new time window for $v_{n-1},\ldots,v_1$.

Step3: denote $e_k = e'_{v1} - t_{v1,0}$, $l_k = l'_{v_1} - t_{v_1,0}$, $[e_k,l_k]$ is the time window for vehicle *K* departing from the depot.

From the step of regression iterative strategy, we can find such fact: for any e'_{v_i} and l'_{v_i} $(v_i \in \{1,2,...,n\})$, they satisfy the following condition: $e'_{v_i} \le l'_{v_i}$. Otherwise, the solution acquired will be the unfeasible solution that violates the time window. In order to improve the feasibility of the strategy, we bring forward a theorem and prove it.

Theorem 4.1: In the new time window, vehicle needn't wait or its' waiting time is zero at any two customers which is adjacent in the same route.

Proof: Assume customer *i* and *j* are serving by vehicle *K*, here *i* is before *j*, that is, vehicle *K* visit *i* early, and their new time windows are $[e'_i, l'_i]$ and $[e'_j, l'_j]$ after we adjust all the time windows. The new time when service at customer *i* and *j* is denoted by b'_i and b'_j. Let w'_i and w'_j be the new waiting time at customer *i* and *j* separately, then,

$e'_i \le b'_i \le l'_i$, $e'_i = \max\{e_i, e'_j - t_{ij} - s_i\}$, $l'_i = \min\{l_i, l'_j - t_{ij} - s_i\}$.

Apparently, it is feasible to visit customer *i* and *j* in the new time window, now we will prove $w'_i \le 0$ and $w'_j \le 0$.

(1) If customer *i* is the first customer that vehicle *K* visits, $w'_i = 0$. Now customer *j* is the second one, and $w_j' = \max\{0, e_j' - (b_i' + s_i + t_{ij})\}$, $e_j' - (b_i' + s_i + t_{ij}) = (e_j' - s_i - t_{ij}) - b_i' \le e_i' - b_i' \le 0$, that is, $w'_j \le 0$.

(2) If customer *i* is the m^{th}, the customer *j* is the next one, and $w'_i \le 0$, $w'_j \le 0$.

So when customer *i* is the m+1th, *j* is the m+2th, $w'_i \le 0$ and $w'_j \le 0$ from that we have just proved.

Summarize from above that we just have proved, when customer *i* is any customer that serviced by vehicle K, there $w'_i \le 0$ and $w'_j \le 0$.

This completes the proof.

5 The Experimental Result

For the VRPTW, there are many publications using heuristics and meta-heuristics, Solomon's 56 benchmark problems are frequently used for the comparison of different heuristics. The test of this work is executed over the type C101 in Solomon's VRPTW instances. Therefore, our algorithm is at least comparable to previous methods in terms of total distance traveled, the control parameters can be found in Table 1.

Table 1. Algorithm Paramenters

T_0	α	m	M	l	L	T'
100	0.8	20	30	5	30	0.8

The best results achieved by our algorithm for VRPTW are compared with the best published results [4]. According to Table 2. Our algorithm generated equivalent solution in C101. Meanwhile, the time that every vehicle departing from each node is given in appendix.

Table 2. Result for the C101

route	the order of the customers visited	distance
1	0-20-24-25-27-29-30-28-26-23-22-21-0	50.80355
2	0-67-65-63-62-74-72-61-64-68-66-69-0	59.40312
3	0-5-3-7-8-10-11-9-6-4-2-1-75-0	59.61804
4	0-43-42-41-40-44-46-45-48-51-50-52-49-47-0	64.80748
5	0-90-87-86-83-82-84-85-88-89-91-0	76.06957
6	0-13-17-18-19-15-16-14-12-0	95.88471
7	0-98-96-95-94-92-93-97-100-99-0	95.94313
8	0-32-33-31-35-37-38-39-36-34-0	97.22717
9	0-57-55-54-53-56-58-60-59-0	127.2975
10	0-81-78-76-71-70-73-77-79-80-0	101.8826
total		828.9369

We can see the comparison in some literatures through Table 3.

Table 3. Comparison in some literatures

Alogrithm	NV	TD	Reference
PFIH	10	853	Solomon M.M.(1983)
ACS	10	833	Xuan Tan et.al (2005)
GIDEON	10	833	Thangiah et.al (1993)
CGH	10	828.94	Alvarenga.G.B.(2004)
mGA	10	828.94	K.C.Tan et. al (2001)
Best Result	10	827.3	Desrochers et.al (1992)
Our algorithm	10	828.937	Jiangwx et.al (2007)

6 Conclusion

In this paper, we propose a two-stage optimization strategy for the Vehicle Routing Problem with Time Windows. The algorithm firstly constructs an initial solution by the stochastic PFIH to provide a superior initial solution, and then optimizes the initial solution using the hybrid algorithm that combines SA and LNS. Finally, a regression iterative strategy is put forward to adjust time windows for the customers and figure out the optimum time for each vehicle departing. It can make the total waiting time zero.

The experimental results have shown that the algorithm greatly improved the solution's quality. When compared with previous approaches, it also suggests the directions for future work exploring the incorporation of additional local search operators into the meta-heuristics.

References

1. Alvarenga, G.B., Mateus, G.R.: Hierarchical Tournament Selection Genetic Algorithm for the vehicle Routing Problem with Time Windows. In: HIS 2004, Fourth International Conference, pp. 410–415 (2004)
2. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35(2), 254–265 (1987)
3. Oliveira, H.C.B., Vasconcelos, G.C., Alvarenga, G.B.: Reducing traveled distance in the vehicle routing problem with time windows using a multi-start simulated annealing. In: IJCNN 2006, pp. 3013–3020, July 16-21 (2006)
4. Alvarenga, G.B., Mateus, G.R.: A two-phase genetic and set partitioning approach for the vehicle routing problem with time windows. In: Proc. 4th International Conference on Hybrid Intelligent Systems, pp. 428–433 (2004)
5. Lim, A., Zhang, X.: A two-stage heuristic for the vehicle routing problem with time windows and a limited number of vehicles. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005), p. 82c (2007)
6. `http://web.cba.neu.edu/~msolomon/problems.htm` (2004)
7. Shaw, P.: Using constraint Programming and local search methods to solve vehicle routing problems. In: Proceeding of the 4th International Conference on Principle and practice of Constrant Programming, pp. 417–431 (1998)

Appendix

The optimum time for each vehicle departing at each customer:

Vehicle1: 21.3944 121.394 216.394 308.394 494 589 682 774 867 960 1052 1062.2

Vehicle2: 20.7934 123 214 306 401 494 589 682 774 867 962.385 1054.39 1070.2

Vehicle3: 21.1933 126.326 217.326 309.326 402.154 495.76 588.76 681.922 774.158 866.394 960 1052 1145 1160.81

Vehicle4: 0.220282 106.773 199.773 291.773 383.773 476.773 569.602 661.602 753.602 846.602 938.838 1032 1125 1217 1235.03

Vehicle5: 24.3845 135 230 321 417 510 605.831 698.659 791.659 884.488 978.093 1000.45

Vehicle6: 20.5979 141.404 236.789 328.789 422.394 516 608 703 798 893 926.54
Vehicle7: 17.9921 139.615 231.615 327 422 517.831 609.831 704.831 799 798 893 926.541
Vehicle8: 24.1807 161.615 254.615 346.615 536.615 629.615 723.615 814.615 910 961.478
Vehicle9: 21.1942 142 236 329 424 519 614 706 799 837.079
Vehicle10: 24.6148 149.615 241.615 336.615 432 618 711 811.44 846.497

An Architecture Design for Evolvable Computer

Xuesong Yan[1,2], Chen Shi[3], Lishan Kang[1], and Shitan Huang[3]

[1] School of Computer Science, China University of Geosciences, Wuhan, China
yanxs1999@126.com
[2] Research Center for Space Science and Technology,
China University of Geosciences, Wuhan, China
[3] Xi'An Microelectronics Technology Institute, Xi'an, China
shichen0528@126.com

Abstract. In recent years, there has been a remarkable growth of the portable computing devices market. To develop and design a scalable, programmable architecture that can be dynamically reconfigured during program execution to a new circuit structure is a key design goal for self-reconfigurable computing systems. In this paper we proposes a new architecture of evolvable computer, this architecture consists of a general-purpose processor, a reconfigurable component (programmable array) and a evolutionary algorithms processor (DSP), and describe the work mechanism of the evolvable computer in detail.

1 Introduction

The computers for deep space exploration to Moon, Mars and beyond faces new challenges, including the extreme environment, the delay of too-long information transportation, the expensive cost of launching and transportation, the unknown situation in space sailing, and so on [1,2]. Environments with large temperature swings is one typical factor induce drifts, degradation, or damage into electronic devices, such as between -180°C and 120°C at the initial landing sites on the Moon, low temperatures of -220°C to -230°C during the polar/crater Moon missions, high temperature of 460 °C for Venus Surface Exploration and Sample Return mission, etc [3].

The current approach for deep space exploration computers designs is to use commercial/military range electronics protected through passive (insulation) or active thermal control, and high weight shielding for radiation reduction. In fact, that only solves problems partly, but adds to sizable weight and volume, compounded by power loss, and additional cost for the mission. More importantly, as missions will target operations with smaller instruments/rovers and operations in areas without solar exposure, these approaches become infeasible. Therefore, we must explore a new technology which is suitable for computers designs for deep space explorations with several advantages including lower costs, less power, higher reliability, higher flexibility and longer life-span, etc. Evolvable hardware (EHW) is an emerging technology with reconfigurability and evolvability suit for space applications. The core of the EHW technology is using advanced search algorithms to automatically design, reconfigure, adapt, or otherwise manipulate hardware or software models of hardware,

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 206–211, 2008.
© Springer-Verlag Berlin Heidelberg 2008

typically evolutionary algorithms (EAs) which are stochastic search methods that mimic the metaphor of natural biological evolution, such as selection, recombination, mutation, migration, locality and neighborhood. Normally EHW is built on software-reconfigurable logic devices, such as the PLD (Programmable Logic Device) and the FPGA (Field Programmable Gate Array).

2 Computing Systems for Deep Space Applications

Reconfigurable computing systems [4] are systems that consist of some reconfigurable hardware along with software programmable processors. The reconfigurable component provides the ability to configure or customize the system for one or more applications. In the ideal case, a reconfigurable system delivers high performance typical of ASIC devices and also provides the flexibility of a general-purpose processor (i.e. it can execute a wide range of applications). Conventionally, field programmable gate arrays (FPGAs) are the most common devices used for implementing reconfigurable components. This is because FPGAs allow designers to manipulate gate-level devices such as flip flops, memory and other logic gates. However, FPGAs have certain disadvantages such as low logic density and inefficient performance for word-level data path operations. Hence, many researchers have proposed prototypes of reconfigurable computing systems that employ non-FPGA reconfigurable components such as DPGA , Garp , PADDI , MATRIX , RaPiD, REMARC , and RAW.

Self-reconfigurable computing systems (an other name of EHW) [5] can changes their architecture and behavior dynamically by themselves while executing a task in order to adapt to their environment and refine their own programming for better performance. They are the most challenging and powerful reconfigurable computing systems. Usually, self-reconfigurable system is implemented on programmable logic devices such as FPGAs. EHW can fall into two categories: extrinsic (or off- line) EHW and intrinsic (or on-line) EHW. In extrinsic EHW, evolution is determined by software, in intrinsic EHW, the hardware controls its own evolution. Currently, EHW has most been studied in terms of off-line adaptation and EHW is a very interesting research field in the artificial intelligence.

Building the electronic system for deep space applications with the EHW technology has already got a lot of achievements, for example “Intelligent Decision-Making for Autonomous Rover Operations” which will be used to explore the Mars in 2009 by NASA[6]. The main object of this research item is to find the best routing strategy even in the unknown environment which sufficiently taking all kinds of factors into account, such as the requirement of exploration missions, the performance of electric systems, the condition of space environment, and so on. These instances of EHW in enabling a fully autonomous space exploration mission have one representative characteristic that is intelligent ability. The main tasks of the intelligent system including two categories: ones that handle the high-level heuristic functions of decision-making and ones that control the low-level autonomic reactive behaviors. For example, during the course of mission operation, it support the planning and scheduling of mission activities to accomplish the overall mission goals and objectives, taking into account the operational status of flight and ground mission elements, health and safety constraints, optimal resource allocation, and science prioritization.

Normally, the architecture of it has multi-level interconnects and interfaces, which offers extreme flexibility and is radically reconfigurable by using both evolvable hardware and intelligent software on the system level as well as the module level. The whole intelligent system is a highly integrated hardware-software system that is analogous to the integration of the neural system with the body in biological systems. In order to design and analyze the system expediently, it can be separated into three basic levels including system level, sub-system level, and Module level. The hierarchy can provide large-scale failsafe redundancy and adaptability to a broad range of operational environment. System level realizes the swarm reconfiguration and reallocation of space system with multi-agent coordination technology. Current reconfigurable space system concepts generally achieve different system configurations by rearrangement of system modules with fixed functional characteristics. And the system-level reconfiguration is allowed to evolve with science priorities and system degradation and attrition. The functional groupings and the operational status of the swarm will be continuously evaluated throughout the mission and the constituents optimized to meet the mission goals. Subsystem level includes an innate set of functional subsystems that are connected by programmable, adaptable and evolvable interfaces to provide functional plasticity.

3 Architecture Design of Evolvable Computer

From the above analyze, we can know in the EHW implementation may be meets some problems. One is the traditional most simple expression design method is no longer the best choice in the hardware evolution, selects the small scale electronic circuit, which has the complexity and more reuse, being the fundamental component, can cause the electronic circuit expression more simple, and is helpful for the hardware realization, improvement the evolution efficiency. The other is for the optimized electronic circuit structure, the evolution tends to use fine-grained granularity component, such easy to constitute the new combination form, obtains the high performance sign electric circuit, uses the coarse-grained granularity component to be advantageous to fast forms the electric circuit which satisfies the function request, but usual performance index insufficient and the fitness value not good.

For these reasons, our evolvable computer architecture comprises three major components: the programmable array, control processor, evolutionary algorithm processor (DSP). Fig.1 shows the organization of the evolvable computing system.

Control processor: Since most target applications involve some sequential processing, a RISC processor, is included in the system. This is a MIPS-like processor with a 4-stage scalar pipeline. It has a 32-bit ALU, register file and an on-chip data cache memory. This processor also coordinates system operation and controls its interface with the external world. This is made possible by addition of specific instructions (besides the standard RISC instructions) to the processor ISA. These instructions initiate data transfers between main memory and our system components, and control execution of the DSP.

Evolutionary algorithm processor (DSP): this component like the traditional DSP, but the only different is evolutionary algorithm will be execute and get the configuration bit-string. So this component is the source of the programmable array.

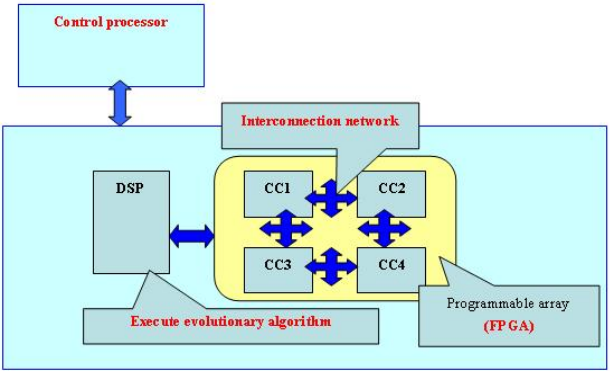


Fig. 1. Architecture of evolvable computer

Programmable array: for this component the most important feature is the interconnection structure is evolvable. So in this component the main work is design the interconnection structure (like Fig.2 show), and uses the basic computing element which has the function to optimize, connection to form the component.

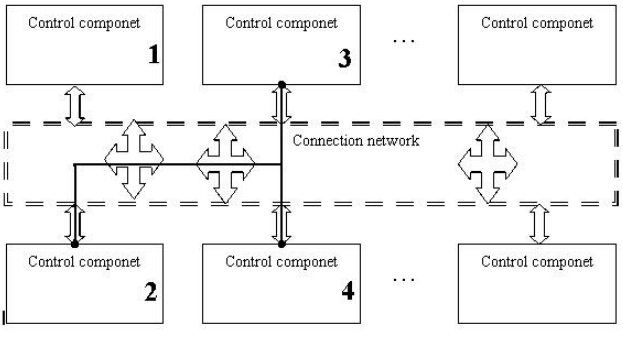


Fig. 2. Interconnection structure

The interconnection structure has two essential features: One is the granularity choice-component level, the processing unit function may establish as same or the different form, but the data operation may use the character level operating mode, guarantees the high performance, for example the 8 or 16 bit data operations. In fine-grained granularity system, processing unit including traditional logical gate, trigger, and search table LUT and so on. They operate with bit, and complete the boolean function of the limited state machine. This structure suits the bit operation and the logic operation. The fine-grained granularity array is usually based on the FPGA system. The FPGA structure has the fine-grained granularity programmable logical block and the programmable internet, provided for the system has been very big the flexibility, when FPGA completed the computation task had following shortcoming: Because uses the bit operation pattern, therefore in realization the operation of wide

data way, the unit maybe including many process unit (bit level).Then the unit interconnection line layout expenses are big, the chip area use rate is low, also the power cost is big. The massive processing unit, the wiring switch and the corresponding disposition data takes the massive resources, simultaneously costs the long disposition time, is unable to realize the dynamic change. The logical level programming causes FPGA the application development to be similar to the VLSI design. Compares with the general processor, the source is difficult in the usual higher order language algorithm mapping on FPGA to realize, because the FPGA granularity and the source code operation width does not match. The massive time use in synthesizing.

In coarse-grained granularity system, processing unit including integrity function unit, for example completes the character operation ALUs/or the multiplier, and more suitable completes the arithmetic operation of character size data. The coarse-grained granularity structure has the multi-bit wide data way and the complex operation operator overcomes based on the FPGA numerical procedure shortcoming. The wide data way may complete on the chip effectively the complex operation, in addition, also may avoid the bit level processing unit constitution complex operator big layout expenses. In processing unit interconnection line aspect, because the connection is the multi-place number according to wide, therefore each line all has the high area use factor, processing unit quantity are less than several magnitudes FPGA, the few segments may reduce the area which the overall situation wiring occupies. The low quantity and the high granularity communication segment have provided the rich correspondence resources, but the fine grain structure is actually the low efficiency. In the practical work both includes the fine-grained granularity operation and the coarse-grained granularity operation, therefore selective the process unit granularity usually needs to adopt compromised. But the mixed granularity unit both to be possible to complete the bit level operation, and may completes the character level operation. For example RAW, CHESS and so on... So we think the structure decided not only by a processing unit granularity, but also by the interconnection network.

For the interconnection structure the second feature is the interconnection network - evolvable: the configuration data may download to the inactive part of the structure, but does not use the operation of the interrupt processing part. Disposition data downloading or the heavy load operates through the internal inserting processor kernel complete. The basic interconnection structure is decided by the processing unit overall situation arrangement way. Simultaneously needs to consider the system aims at the application domain and completes complexity. Very obvious, the correspondence structure type affects directly the application mapping complexity. At present, the most main structure is two-dimensional Mesh structure. In some other structures, the processing unit is arranged or many linear arrays. In the third structure, connects each processing unit with crossbar, like this may realize connects willfully.

Mesh structure: This structure will array the processing unit according to the rectangular, if the simple interconnection way horizontal and vertical all around unit will be connected for each unit. This kind of structure may effectively realize parallel and the full use correspondence resources. But must use the grid fully the superiority, must carry on the reasonable effective layout and the arrangement to the data. This step has the very tremendous influence to the application execution, around the processing unit is close to the unit is the direct correspondence resources. Linear array structure: This structure processing unit by the emissions is or many linear arrays, between usual neighboring unit interconnection. This kind of organization mainly

uses for to map the assembly line algorithm, each processing Yuan is equal to the assembly line operation. The linear array suits in does not have cross's application. Full crossbar connection: The permission processing unit the willfully connection, is the most formidable correspondence network. This enables the wiring duty to become the simple operation. However, because completes full the crossbar connection to need the very high price, at present this kind of structure system only completes a simple level connection. The three structure have their own feature in our design we want to use the three structure mixed together, then the interconnection structure will be more flexible.

4 Conclusion

With the EHW technology developing, the hardware of self-reconfigurable computing systems becomes EHW and the software becomes corresponding intelligent software that forms a self-organize, self-diagnose, and self-recover computer system. In the early days of the EHW research, we aim at the optimize circuit design technology of the electronic system including evolvable computer design with the EAs. In this paper we proposes a new architecture of evolvable computer, this architecture consists of a general-purpose processor, a reconfigurable component (programmable array) and an evolutionary algorithms processor (DSP). For this architecture, we have described about the mechanism in detail, the important feature of this architecture is evolvable. In this architecture, the important is the interconnection structure design, in the structure we emphasize on the granularity selection and the interconnection network selection. The next work is implementation this architecture with FPGAs and compared the performance with other reconfigurable computing systems.

Acknowledgements

This paper is supported by Astronautics Research Foundation of China (NO. C5220060318).

References

[1] Von Neumann, J.: The Theory of Self-Reproducing Automaton. University Illinois Press, Urbana (1966)

[2] de Garis, H.: Evolvable Hardware: The Genetic Programming of Darwin Machines. In: Proc. of Int. Conf. on Artificial Neural Nets and Genetic Algorithms, Innsbruck, pp. 441–449 (1993)

[3] Proceedings of the NASA/JPL Conference on Electronics for Extreme Environments, Pasadena, CA, February 9-11 (1999), http://extremeelectronics.jpl.nasa.gov/conference

[4] Lee, M.-H., Singh, H., Lu, G., et al.: Design and Implementation of the MorphoSys Reconfigurable Computing Processor. Journal of VLSI Signal Processing Systems, 147–164 (2000)

[5] Lu, G.: Modeling, implementation and scalability of the MorphoSys dynamically reconfigurable computing architecture, PhD thesis (2000)

[6] Shi, C., Huang, S., Yan, X.: The Design of an Evolvable On-Board Computer. In: Kang, L., Liu, Y., Zeng, S. (eds.) ICES 2007. LNCS, vol. 4684, pp. 292–296. Springer, Heidelberg (2007)

Gene Expression Programming Neural Network for Regression and Classification

Weihong Wang, Qu Li, and Xing Qi

Software College, Zhejiang University of Technology,
Hangzhou, China, 310032
wwh@zjut.edu.cn

Abstract. Gene Expression Programming(GEP) is a kind of heuristic method based on evolutionary computation theory. Basic GEP method has been proved to be powerful in symbolic regression and data mining tasks. However, GEP's potential for neural network learning has not been well studied. In this paper, we prove that basic GEP neural network(GEPNN) is unable to solve difficult regression and classification problems. Based on our proof, we propose an extended method for evolving neural network with GEP. The extended GEPNN is used in function finding and classification problems. Results on multiple learning methods show the effectiveness of our method.

1 Introduction

Gene expression programming[1][2] is a kind of evolution computation method based on genetic algorithms and genetic programming. It shares the fundamental evolutionary idea of evolution computation method and uses genotype/phenotype mapping to overcome difficulties of GA and GP. GEP has been proved to be versatile and robust on various learning tasks such as symbolic regression, cellular automata, classification and association rules mining and time series analysis[1][2][3].

Artificial neural network(ANN)[4] is a kind of learning method which has been well studied in artificial intelligence field. An ANN is a computational device consists of many simple connected units called neurons which work in parallel. The connections between the units or nodes are usually weighted by real-valued weights.

Researchers have tried many evolution computation methods to evolve the topology and weights and thresholds of the networks[5]. Among all these theories, the most important and common method of neural network training is to fine-tune the weights and thresholds of the networks.

Although GA has been used in neural network evolution by many researchers, GP is not commonly used for the same task. Ritchie et al. developed a genetic programming optimized neural network (GPNN)[6] and used it to improve the identification of genetic and gene–environment combinations associated with disease risk. As GP did on NN, GEPNN[2] was developed in an attempt to improve upon the trial-and-error process of choosing an optimal architecture for a pure feed-forward back propagation neural network. Since the difference between GEP and GP has been well studied in references[1][2], the superiority of GEPNN against GPNN is almost self-evident.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 212–219, 2008.
© Springer-Verlag Berlin Heidelberg 2008

GEP has been used to encode and evolve ANN, and GEPNN was used to solve 6-multiplexer and XOR problems. But the original version of GEPNN by C.Ferreira[2] was unable to solve high order regression problems. In this paper, we propose an extended method for evolving neural network with GEP, our method is more natural and robust. The extended method is able to solve high order regression problem with high accuracy and it is also used to solve classification problems.

The rest of this paper is organized as follows. In section 2, we briefly describe the mechanism and components of GEP algorithm. In section 3, we describe the basic GEPNN algorithm and show its limitations with proof. In section 4, we propose our extension and showed its better performance with extensive experiments. Conclusions and future work are given in section 5.

2 Gene Expression Programming Algorithm

Gene Expression Programming[1] is a kind of evolutionary algorithm. It was first invented by C.Ferreira in 2001. It shares the fundamental evolutionary idea of evolution computation method. Unlike GA's fix binary string and GP's parse tree representations, it uses fix length strings as its genotypes. The genotypes are later expressed as phenotypes, i.e., Expression Trees (ETs) with different sizes and shapes. Usually five components are specified when using GEP to solve a practical problem: the function set, the terminal set that includes problem-specific variables and pre-selected constants, fitness function, control parameters, and stop condition.

In GEP, a chromosome is composed of predefined number of genes. Each gene is divided into head and tail parts. The size of the head (h) is determined by the user, and the size of the tail (t) is computed as: $t = h(n - 1) + 1$, considering n the maximum arity in the function set for the particular problem. The part not included in the expression tree is called non-coding area. For example, a GEP gene with the head length of 6 could be as follows:

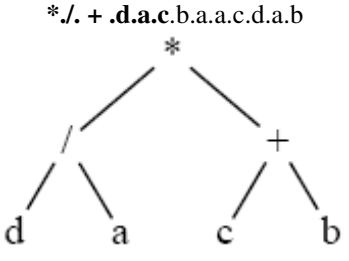


Fig. 1. A GEP Expression tree

When multiple genes in a chromosome are expressed, they are connected by linking functions such as arithmetical functions or Boolean functions. By this way, several ETs can be combined to form a bigger ET. For detailed description of GEP. Please refer to[2].

3 GEP Neural Network

The GEPNN[2] proposed by C.Ferreira uses the basic GEP coding method and evolution strategy as its fundamental. The GEPNN optimizes the inputs from a larger pool of variables, the weights and the connectivity of the network including the number of hidden layers and the number of nodes in the hidden layer. Thus, the algorithm attempts to generate optimal neural network architecture for a given data set. This is an advantage over the traditional back propagation NN(BPNN) in which the inputs and architecture are pre-specified and only the weights are optimized.

As C.Ferreira described in[2]: Structurally, a neural network has three different classes of units: input units, hidden units, and output units. An activation pattern is presented on its input units and spreads in a forward direction from the input units through one or more layers of hidden units to the output units. The activation coming into a unit from other units is multiplied by the weights on the links over which it spreads. All incoming activation is then added together and the unit becomes activated only if the incoming result is above the unit's threshold.

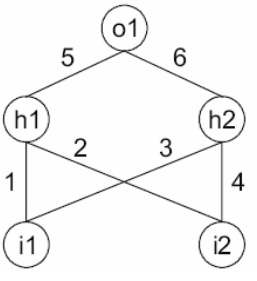


Fig. 2. A neural network topology

Consider, for instance, in Fig 2, the conventionally represented neural network with two input units (i_1 and i_2), two hidden units (h_1 and h_2), and one output unit (o_1) (for simplicity, the thresholds are all equal to one and are omitted).

It can also be represented by a conventional tree:

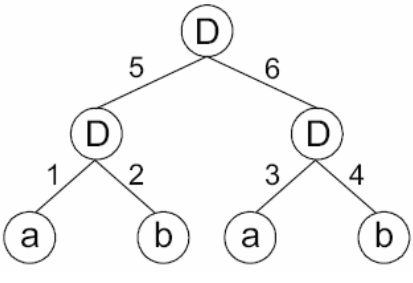


Fig. 3. Conventional tree representation of ANN

where a and b represent, respectively, the two inputs i_1 and i_2 and "D" represents a function with connectivity two. This function multiplies the values of the arguments by the respective weights and adds all the incoming activation in order to determine the forwarded output. This output (zero or one) depends on the threshold, that is, if the total incoming activation is equal to or greater than the threshold, then the output is one, zero otherwise.

We could linearize the above NN-tree as follows:

```
0123456789012
DDDabab123456
```

where the structure in bold (Dw) encodes the weights. The values of each weight are kept in an array and are retrieved as necessary.

4 GEP Neural Network Extension

Although GEPNN has been used to solve two important problems, i.e., XOR problem and multiplexer problem, its power in symbolic regression has not been well studied. Basic GEPNN can not been used to solve regression problems harder than one dimension function finding. Because default function set of basic GEPNN UDTQ are all linear functions(where "U" represents a function of one argument, "D" represents a function with two arguments, "T" represents a function of three arguments, and "Q" represents a function of four arguments.). It is unable to model high order functions other than linear function such as y=ax+b.

By analyzing the coding method of basic GEPNN, the most natural and effective way to solve above problem is to add functions other than linear function such as UDTQ, for example, multiply and division, which are often used but not included in GEPNN. Their meanings are just as they often used in ordinary calculations.

For example, if we have a chromosome with h = 8 in the genotype, its corresponding genotype is:

```
*-+aaaa469818617
```

Its corresponding expression tree is(the part not included in the expression tree is non-coding area, the bold numbers are domain numbers for weights):

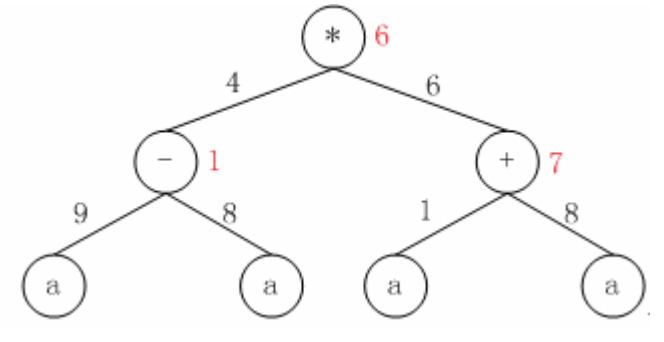


Fig. 4. A GEPNN expression tree

with their weights are: {-1.553,1.612,-1.606,-0.487,1.475,-0.253,-1.91,1.427,-0.103,-1.375}and their thresholds are: {-1.14,1.125,-1.173,-0.74,0.393,1.135,-0.625,1.523,-0.029,-1.634}. Its expression tree is:

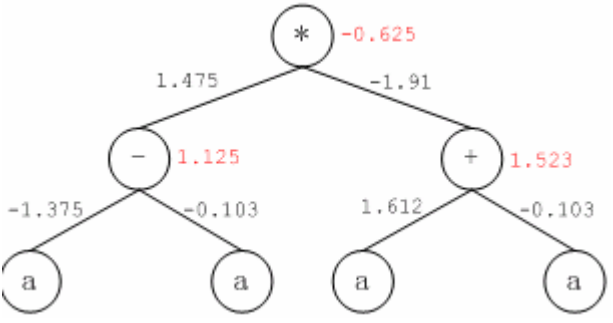


Fig. 5. A GEPNN expression tree with explicit weights and thresholds

By this way, we can regard GEPNN as a special kind of basic GEP, i.e., A GEPNN expression tree is a special case of basic GEP expression tree with weight and thresholds.

5 Experiments and Results

5.1 GEPNN for Regression

We implemented the GEPNN in WEKA platform[8], a well recognized machine learning toolkit package. In experiment I, we use the polynomial function of $y = 2.718a^2 + 3.146a$, which is a relatively difficult problem among many function finding tasks. We run the experiment 5 times. The parameters of each run are given in table 1.

Table 1. Parameters for all experiments

Number of runs	10	Chromosome length	41
Max generation	5000	Mutation rate	0.044
Population size	20	One/Two point crossover rate	0.6
Function set	*+/	IS transposition rate	0.1
Terminal set	a	IS element size	1,2,3
Gene number	1	RIS transposition rate	0.1
Head length	8	RIS element size	1,2,3
Max arity	2	Extra domain transposition rate	0.1

In experiment I, we use the absolute error fitness function:

$$f_i = \sum_{i=1}^{C_t} \left(M - \left| C_{(i,j)} - T_j \right| \right) \tag{1}$$

where M is the range of selection, $C_{(i,j)}$ the value returned by the individual chromosome i for fitness case j (out of C_t fitness cases) and T_j is the target value for fitness case j. If $C_{(i,j)} - T_j|$ (the precision) less or equal to 0.01, then the precision is equal to zero, and $f_i = f_{max} = C_t$. M For this problem, we will use an M = 100 and, therefore, f_{max} = 1000 for training and f_{max} =800 for 5-fold cross-validation.

The result of experiment for regression is given in Table 2. It shows that the GEPNN can find good approximation to the polynomial function.

Table 2. Experiment result

Runs		1	2	3	4	5
Training Fit.		994.715	998.073	987.709	998.920	999.102
5-fold C.V.	1	796.616	774.495	792.688	789.612	795.688
	2	783.504	795.907	793.976	771.587	786.518
	3	798.854	795.046	761.942	770.827	781.870
	4	788.964	794.003	787.347	792.523	789.859
	5	795.619	795.938	797.364	785.196	769.518

5.2 GEPNN for Classification

In classification learning tasks, we solved eight classification problems from UCI data repository[7] with GEPNN, the features of the eight datasets are given in table 3.

Table 3. Characteristics of the Benchmark Data Sets

Database	No. Cases	No. of Attr.	No. Class
Balance Scale	625	4	3
Breast Cancer W	683	9	2
Glass	214	9	6
Iris	150	4	3
Labor	57	15	2
Pima Indian	768	8	2
Wine	178	13	3
Zoo	101	17	7

Problems given in above table show various kinds of complexity, with numbers of fitness cases from tens to hundreds, number of classes from two to seven, number of attributes from four to seventeen.

We implemented our GEPNN on classification with the strategy of round threshold as GEP does. i.e., for the 0/1 rounding threshold converts the output of an individual program into 1 if the output is equal to or greater than 0, or into 0 otherwise. For multiple classes we adopt the often-used *one-against-all* learning strategy to transform the an n-class problem into n 2-class problems. We use the MultiClassClassifier in weka.classifier.meta package as meta classifier. MultiClassClassifier also solves multiple class problems with one-against-all strategy.

The results of classification accuracy with standard GEP and our GEPNN are given in table 4.From table 4, we can see that on the eight data sets we used in the

experiment, GEPNN won on 4 data sets and lose on 4 datasets. That means GEPNN's can not only be used to solve classification problems but also can get almost the same performance on classification as basic GEP classification method does.

Table 4. Classification result

Database	GEP	GEPNN
Balance Scale	**91.58**	88.38
Breast Cancer W	95.65	**95.91**
Glass	**60.47**	57.94
Iris	**95.6**	94.00
Labor	64.91	**75.79**
Pima Indian	69.67	**72.32**
Wine	**90.79**	89.33
Zoo	87.53	**89.31**

6 Conclusion

In this paper, we presented a hybrid GEPNN which utilize the linear representation of GEP. We used basic arithmetic operators to overcome the limitation of basic GEPNN proposed by C.Ferreira.[2]. Results showed that our approach can solve difficult symbolic regression problem as well as various kinds of classification problems. The performance of our approach was fairly satisfactory with respect to its moderate accuracy. However, since this is only a preliminary research of GEPNN for regression and classification, the accuracy can still be improved. On the other hand, the coding method of GEPNN is much more complicated than that of basic GEP, the searching space is much bigger than basic GEP. In the future, we will focus on how to narrow the problem of searching space.

Acknowledgement

This work is supported by Science and Technology Ministry Foundation of Zhejiang Province under Grant 2004C23034, Natural Science Foundation of Zhejiang Province under Grant Z105391 and Y107551, National 863 Project under Grant 2006AA01Z235, Education Ministry of Zhejiang Province under Grant 20060815.

References

1. Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems (2001)
2. Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, 2nd edn. Springer, Germany (2006)

3. Zhou, C., Xiao, W., Nelson, P.C., Tirpak, T.M.: Evolving Accurate and Compact Classification Rules with Gene Expression Programming. IEEE Transactions on Evolutionary Computation 7(6), 519–531 (2003)
4. Haykin, S.: Neural Networks: A Comprehensive Foundation. IEEE Society Press, Macmillan College Publishing (1992)
5. Ritchie, M.D., Coffey, C.S., Moore, J.H.: Genetic Programming Neural Networks as a Bioinformatics Tool for Human Genetics. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 438–448. Springer, Heidelberg (2004)
6. Yao, X.: Evolving artificial neural networks. Proc. IEEE 87(9), 1423–1447 (1999)
7. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases. Univ. California at Irvine, Dept. Inform. Comput. Sci., CA (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
8. Witten, I.H., Frank, E.: Data Mining Practical Machine Learning Tools and Techniques with java Implementations. Morgan Kaufmann Publishers, USA (2000)

Granular-Based Linguistic Models for Identification of Process System

Keun-Chang Kwak

Dept. of Control, Instrumentation, and Robot Engneering, Chosun University
Seosuk-dong Dong-gu, Gwangju, 501-759, Korea
kwak@chosun.ac.kr

Abstract. This paper is concerned with a method for designing a Granular-based Linguistic Model (GLM) for identification of process system. For this purpose, a GLM is constructed by the use of fuzzy granulation realized via linguistic context-based fuzzy clustering. The proposed approach is comprised of three steps. In the first step, we use Linear Regression (LR) for global approximation. As a result, we obtain the approximation error. On the basis of the error, we construct the GLM based on the linguistic contexts produced by the error distribution for local approximation in the second step. Finally we use the Constrained Least Square Estimate (CLSE) method to generate the flexible linguistic context in the consequent part to improve the approximation performance. Finally, we use the proposed methods to identify a pH neutralization process in a continuous stirred-tank reactor (CSTR). The experimental results reveal that the proposed approach yields a better performance in comparison with linear model, Linguistic Models (LM), and GLM itself.

Keywords: Granular-based linguistic models, fuzzy granulation, pH neutralization process, constrained least square estimate.

1 Introduction

Over the past few decades, a considerable number of studies have been conducted on the identification of nonlinear system using fuzzy models [5]. In general, the fuzzy models are divided into three types depending on the particular structure of the consequent part; linguistic fuzzy model [6], fuzzy relational model [7], and Takagi-Sugeno (TS) fuzzy model [9]. In the linguistic fuzzy model, Mamdani [6] model was proposed as the first attempt to control a steam engine and boiler combination by a set of linguistic control rules obtained from experienced human operators. Fuzzy relational model introduced by Pedrycz [7] is constructed by the particular antecedent proposition to be associated with seven different consequent propositions via a fuzzy relation. On the other hand, TS fuzzy model is designed by a systematic approach to generating fuzzy rules from a given input-output data set.

Meanwhile, we develop a Granular-based Linguistic Model (GLM) constructed by the use of fuzzy granulation realized through linguistic context-based fuzzy clustering. For this purpose, we develop the improved architecture based on conventional

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 220–227, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Linguistic Model (LM) proposed by Pedrycz [8]. Although the effectiveness of this model has demonstrated, this model has a poor approximation and generalization capability. To solve this problem, we employ the Constrained Least Square Estimate (CLSE) method [4] to estimate the consequent parameter. Furthermore, we compensate a biased error by adding bias term to a GLM. Thus, we deal with global and local approximation based on the proposed GLM. The proposed method is constructed by combining Linear Regression (LR) for global model with GLM for local model. Finally, we apply to a pH neutralization process in a continuous stirred-tank reactor (CSTR) [1], [3]. The results produced by the proposed method show a better approximation and generalization capability in comparison with linear ARX model, LM, and GLM itself.

This paper is organized as follows. Section 2 describes the architecture and context-based fuzzy clustering for GLM. In Section 3, we present the design method of the proposed LR-GLM. The proposed model is applied to CSTR in Section 4. Finally, conclusion is given in Section 5.

2 Granular-Based Linguistic Model (GLM)

This section describes the architecture of a GLM designed by context-based fuzzy clustering. This clustering method is an effective approach to estimate the cluster centers preserving homogeneity based on the concept of fuzzy granulation. In contrast to the context-free clustering methods such as Fuzzy C-Means (FCM) [2], the context-based clustering method is performed with the aid of the linguistic contexts obtained in output domain. Fig. 1 shows the architecture of the proposed GLM.

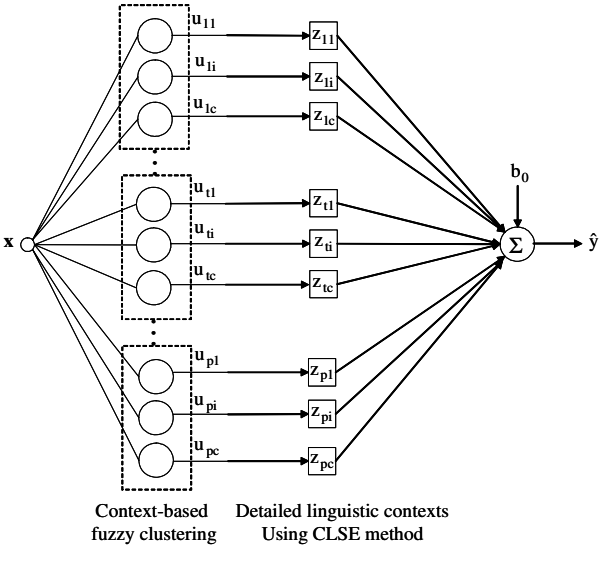


Fig. 1. Architecture of the proposed GLM

The optimization completed by this clustering is realized iteratively by updating the partition matrix and the prototypes. When applying this clustering method to numerical input-output data pairs, each of the cluster centers presents a prototype that exhibits certain characteristics of the system to be modeled. The update of the partition matrix is completed as follows

$$u_{ik} = \frac{f_k}{\sum_{j=1}^{c} \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|} \right)^{\frac{2}{m-1}}} \tag{1}$$

$$i = 1,2,\ldots,c, \quad k = 1,2,\ldots,N$$

where $m \in [1,\infty)$ is a weighting exponent or fuzzification factor. The linguistic contexts to obtain f_k are generated through a series of triangular membership functions with probabilistic distribution along the domain of an output variable and an 1/2 overlap between successive fuzzy sets. The prototypes are calculated in the form

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^m} \tag{2}$$

The computations of the prototypes are the same as for the original FCM clustering algorithm. Moreover, the convergence conditions for the method are the same as thoroughly discussed for the original FCM clustering algorithm [2]. On the other hand, the consequent part is obtained by CLSE method as follows

$$\min_{\theta} \|U\theta - Y\| \quad \text{subject to} \quad \min(Y) \le \theta \le \max(Y) \tag{3}$$

where U and Y denote the activation levels in layer 2 and the actual output, respectively. The parameter θ to be estimated is the modal values of the detailed linguistic contexts. For further details on the CLSE method, see [1],[3].

3 Proposed Approach: LR-GLM

Based on the GLM described in Section 2, we propose two approaches to improve the performance related to global and local approximation. Firstly, let us consider the block diagram of LR-GLM shown in Fig. 2. The LR and GLM are used for global and local model, respectively. The LR is designed to predict a system's behavior for target system. The model can be expressed as follows

$$\hat{y}_L = a_0 + a_1 \mathbf{x_1} + \cdots + a_j \mathbf{x_j} \tag{4}$$

where $\{a_0, a_1, \ldots, a_j\}$ are modifiable linear parameters. $\hat{y}_L$ is the predicted output obtained by LR. The optimal values of these linear parameters were obtained directly by the least-squares method. The error obtained by LR is used as the output in the constructed GLM. In other words, a GLM is constructed to identify the local area represented by approximation error. Thus, the final RMSE (Root Mean Square Error) is calculated by the following form

$$\mathrm{RMSE}_1 = \sqrt{\frac{\sum_{k=1}^{N} \left((\hat{y}_L - \hat{y}_G) - y\right)^2}{N}} \tag{5}$$

where $\hat{y}_G$ and y are the predicted error and the actual output, respectively.

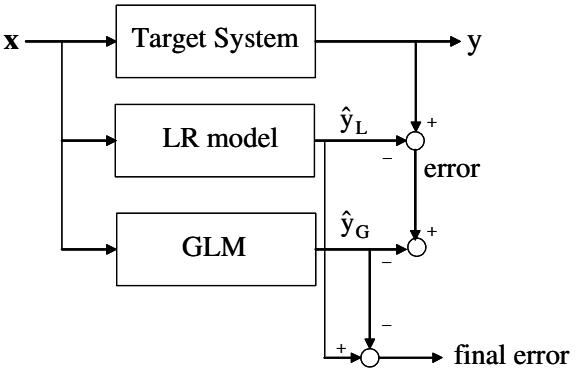


Fig. 2. Block diagram of LR-GLM

4 Experimental Results

We use the well-known benchmark problem with nonlinear dynamics for pH neutralization process in a continuous stirred-tank reactor (CSTR). The input variables are pH(k) and $F_{NaOH}(k)$ in the steady-state process. The output variable to be predicted is pH(k+1). The experimental data set was produced by randomly generating $F_{NaOH}(k)$ in the range of 513-525 l/min. For further details of this dynamic model for a pH in a CSTR, see [1],[3]. The dataset includes 2500 input-output pairs. The random split into the training and testing part is the one of 60%-40%. The experiment was repeated 10 times (10 fold cross-validation). The training data set is used for model construction, while the test set is used for model validation. Thus, the resultant model is not biased toward the training data set and it is likely to have a better generalization capacity to new data. Fig. 3 visualizes the distribution of input data and cluster centers in each linguistic context estimated by context-based fuzzy clustering, respectively. Fig. 4 shows the randomly generated $F_{NaOH}(k)$ in the range of 513-525 l/min.

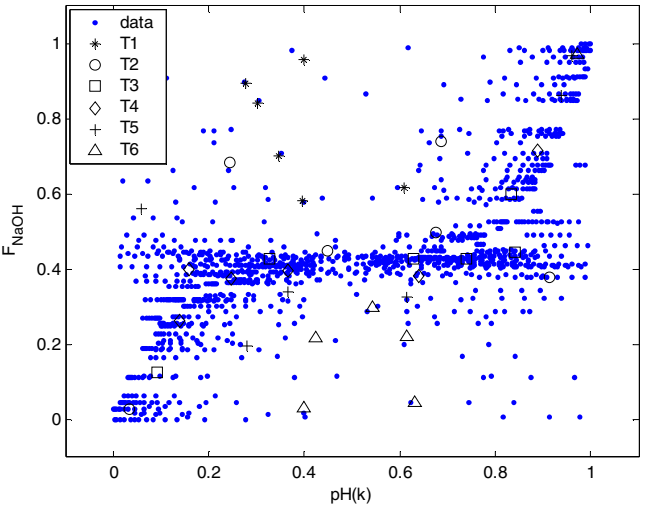


Fig. 3. Data distribution and the estimated cluster centers

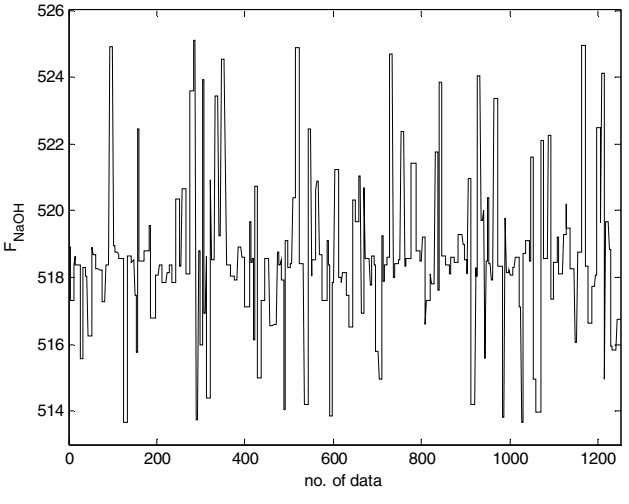


Fig. 4. Randomly generating $F_{NaOH}(k)$ data

In the conventional LM, these contexts were generated through a series of triangular membership functions with equally spaced along the domain of an output variable. However, we may encounter a data scarcity problem due to small data included in some linguistic context. Thus, this problem brings about the difficulty to

obtain fuzzy rules from the CFCM clustering. Therefore, we use probabilistic distribution of output variable to produce the flexible linguistic contexts. Fig. 5 shows the automatic generation of linguistic contexts based on histogram, Probability Density Function (PDF), and Conditional Density Function (CDF). Fig. 6 shows the global approximation error obtained by LR method and local predicted error obtained by GLR method, respectively.

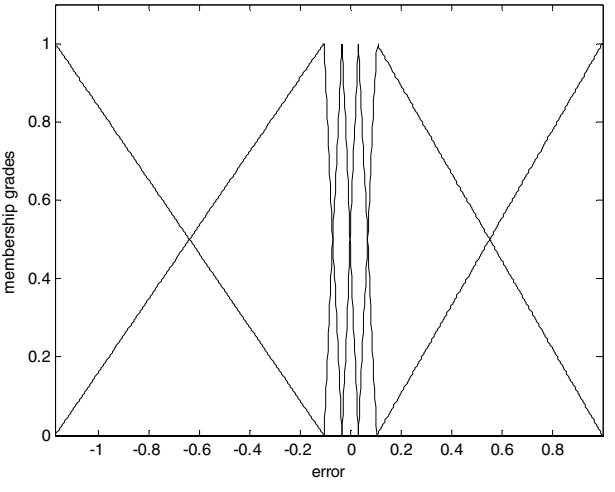


Fig. 5. Generation of linguistic contexts (p=6)

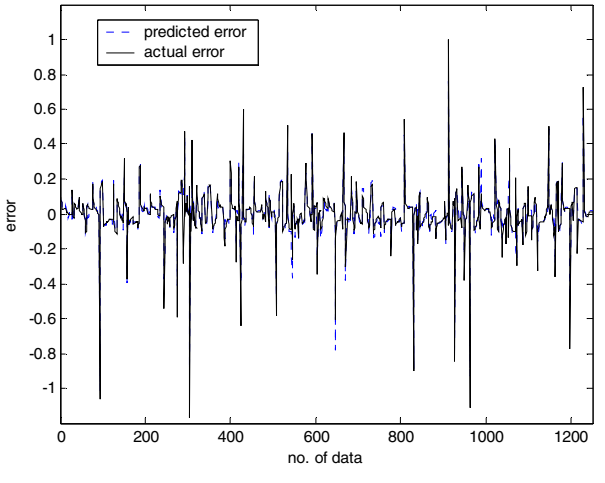


Fig. 6. Global approximation error and local predicted error

We obtained the best performance in case of p=6 and c=6 as the number of “p” and ”c” increase from 2 to 6. Fig. 7 and 8 visualize the prediction performance and error of actual output and model output obtained by the proposed method for test data (one among 10 runs), respectively. The comparative analysis covered in Table 1, shows that the LR-GLM method yields better performance in compared with linear ARX model, conventional linguistic model, and GLM itself.

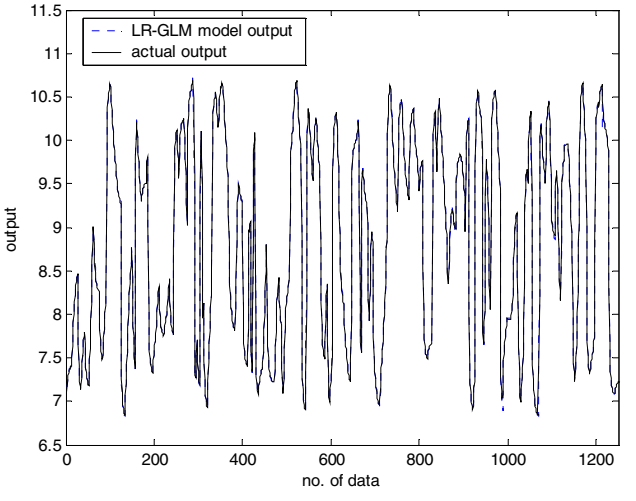


Fig. 7. Identification performance obtained by LR-GLM approach

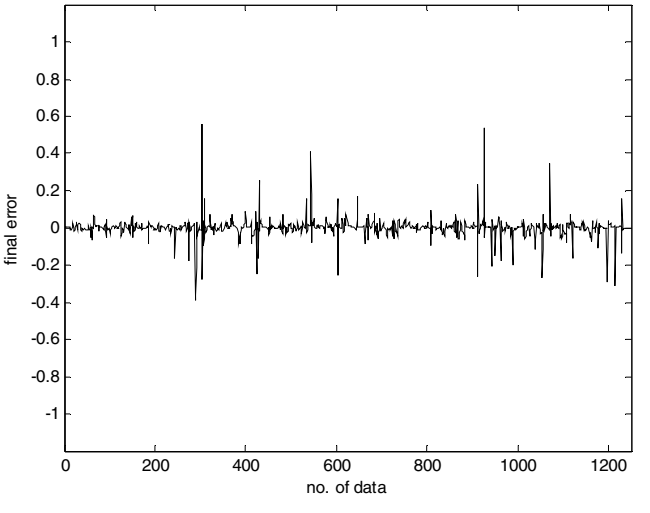


Fig. 8. Prediction error obtained by LR-GLM approach

Table 1. Comparison of RMSE

	p, c	RMSE (Training data)	RMSE (Test data)
Linear ARX model	.	0.547	0.532
LM	p=c=6	0.151 ± 0.009	0.158 ± 0.011
GLM	p=c=6	0.118 ± 0.015	0.115 ± 0.010
Proposed method (LR-GLM)	p=c=6	0.056 ± 0.002	0.067 ± 0.006

5 Conclusions

We have developed the LR-GLM based on the combination of global model and local model. We used the CLSE method to generate the detailed linguistic context in the consequent. Furthermore, we used probabilistic distribution of output variable to produce the flexible linguistic contexts, because the conventional method brings about the difficulty to obtain fuzzy rules from the context-based fuzzy clustering. The experimental results clearly demonstrated that the proposed approach yielded a better performance in comparison with linear ARX model, linguistic model, and GLM itself in conjunction with identification of a pH neutralization process in a CSTR.

References

1. Abonyi, J., Babuska, R., Szeifert, F.: Fuzzy modeling with multivariate membership functions: gray-box identification and control design. IEEE Trans. on Systems, Man, and Cybernectics-Part B 31(5), 755–767 (2001)
2. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
3. Bhat, N., McAvoy, T.: Determining model structure for neural models by network stripping. Computers and Chemical Engineering 16, 271–281 (1992)
4. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1981)
5. Kwak, K.C., Pedrycz, W., Chun, M.G.: Modeling nonlinear systems: An approach of boosted linguistic models. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS, vol. 3614, pp. 514–523. Springer, Heidelberg (2005)
6. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)
7. Pedrycz, W.: An identification algorithm in fuzzy relational system. Fuzzy Sets and Systems 13, 153–167 (1984)
8. Pedrycz, W., Vasilakos, A.V.: Linguistic models and linguistic modeling. IEEE Trans. on Systems, Man, and Cybernetics-Part C 29(6), 745–757 (1999)
9. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. on Systems, Man, and Cybernetics 15, 116–132 (1985)

Particle Swarm Optimization Applied to Restoration of Electrical Energy Distribution Systems

Germano Lambert-Torres, Helga Gonzaga Martins, Maurilio Pereira Coutinho, Camila Paes Salomon, and Leonardo Schilling Filgueiras

Federal University of Itajuba, Av. BPS, 1303, Pinheirinho,
37.500-503, Itajuba, MG, Brazil
{helgagonzaga,germanoltorres,maurilio.coutinho,
leonardoschilling}@gmail.com,
camsalomon@yahoo.com.br

Abstract. This paper presents a technical application of the Particule Swarm Optimization - PSO technique to a reconfiguration problem of a electrical energy distribution system. The proposed methodology consists of the use of the maximization function of the number of loads supplied and the loss minimization by the expansion of the original PSO. The approach utilized the Distribution_System_01. A number of tests were carried out in the system to simulate several fault occurrences in the electrical energy transmission lines. The PSO algorithm encountered the optimal solution in a reasonable CPU time, compared to the dimension of the distribution system.

Keywords: Particle Swarm Optimization, Swarm Intelligence, Electrical Power Systems, Distribution Systems Restoration, Distribution System.

1 Introduction

Electrical Power Systems have been presenting a natural growth due to the increase in demand and consumption of electrical energy. This happens because of modern society's greater dependence on power supply. This situation is more evident in large urban centers and in regions of greater industrial concentration. The development of the system increases the complexity of its supervision, control, and energy demand supply management [1-3]. The mentioned factors highlight the importance of high quality power supply as well as its continuity, low cost and reliability [1], [3].

The possibility of faults along the line is inherent to the system or even greater due to the rise in electrical system complexity and natural factors [1]. Thus, after fault occurrences, it is extremely important that the electric power system restoration be quick to guarantee the power demand supply and the customer's satisfaction. The longer it takes, the greater the loss for the company as well as for the customer. This situation becomes worse when the fault reaches an industrial area [4-5]. The reconfiguration is a switch shifting (open/closed), loss reduction, load balancing, and restoration process [6].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 228–238, 2008.
© Springer-Verlag Berlin Heidelberg 2008

New technologies have been developed aiming at the operation of energy transmission and distribution systems in large scale. These are related mainly to automation, protection and control. New technologies targets computational time efficiency, and are supplied with a certain degree of intelligence implemented through techniques that belong to the Artificial Intelligence area, such as the use Genetic Algorithms-supported modeling - GA [7], [12], Particle Swarm Optimization – PSO [1], [8], [15], [17-18], Ant Colony Optimization – ACO [9], Neural Network - NN [10], Expert Systems - ES [11-12], and Non Classical Logics [3], [13]. This has given their users an immeasurable advance in different control applications seen in the electrical sector [14]. In recent years, Computational Swarm Intelligence has been drawing the attention of researchers and showing special connection with evolutionary strategies [15]. With the improvements carried out in Swarm Intelligence theory and its applications, researchers are using it to solve practical optimization problems. This has become an excellent way towards the solution to optimization- combined problems 16].

This paper proposes the application of PSO as an optimization technique to the restoration of distribution systems. The intended reconfiguration is an optimization and decision-making process which considers the maximization of the number of loads supplied associated to the minimization of the numbers of closed switches. PSO is based on the behavior of animal groups which present some iteration and learning capacity, in such a way that it relates the individual development compared to the group or population [15-16-17-18]. The distribution system tested was the Distribution_System_01 [7].

2 Restoration of Electrical Distribution Systems

The electric energy distribution system presents a radially topological structure. In this kind of system, if all the switches are closed, there will be formation of meshes. For a system with k meshes, k switches have to be open in order to preserve the radial structure of the topology; where only one switch must be open in each mesh, that is, there must be at least one normally open switch in each mesh. Thus, the reconfiguration of the system is done by closing these normally open switches in order to restore the system in an optimized way; however it retains the radial topology [19]. The present approach supposes that for each line there must be one switch, and if in the initial configuration it is not on, then, its normal operational status is open. For an optimized reconfiguration, the decision-making involving the switches, which must be closed, is done so as to maximize the supplied loads and minimize the number of closed switches [19]. Processing must happen fast and dynamically. It should provide a simple, objective, and efficient solution to the operator [1], [3]. Considering fault occurrence on a single line of the distribution system, with the characteristics mentioned above, the closing of a single switch is enough to restore the system. This fact generates a new way for energy transmission, to supply the greatest possible number of load affected by the fault. A peculiarity occurs in case the faulted line is of the kind which has normally open switches, because fault occurrence on the line does not jeopardize the distribution of the system, therefore, the loads supply. In this case the only solution is repairing it, providing necessary technical support.

3 Particle Swarm Optimization

Swarm Intelligence is an optimization technique based on social behavior. It was developed by James Kennedy and Russel Eberhart in 1995 [17-18]. Although the particle swarm optimization principle is relatively new, J. Kennedy and R. Eberhart's predecessor had already thought of alternative ways for application in biological research and animal swarm simulation [20]. The first computational work reported in this research area was an article about the simulation of bird flocks intended to simulate real flocks for application in movies and graph design, published by Craig Reynolds [20]. This led to the development of a group of individual governed by three main rules: separation, alignment and cohesion. Later, Frank Heppner worked with Reynolds' article in 1990, in order to obtain more detailed bird flocks to be applied in animation [21]. In mid 1990s, Jesper Hoffmeyer, a biologist, conceived one of the best definitions of swarms in terms of algorithm: a set of mobile agents presenting the ability to communicate to each other directly or indirectly in their environment and converge collectively for the solution of a given problem [22]. The research on the computational application of PSO actually started with James Kennedy and Russel Eberhart's publication in 2001 [17-18], [20]. Since then, much has been published about the subject, with applications in several areas like function optimization, electric power systems, the traveling salesman problem, telecommunications, among others.

3.1 The PSO Algorithm Technique

PSO is an optimization algorithm based on a population of individuals that present the ability to interact among each other as well as with the environment. Since the individuals are social, they also have the knowledge about their neighbor's behavior and accomplishments in such a way that there exists individual learning and cultural transference [17-18]. The cultural adaptation process is based on three fundamental principles: assessment, comparison, and imitation. The individuals assess their own behavior according to the environment, then compare themselves with the others and imitate the individuals who are better [17-18]. This is how the convergence of the group into a unique solution of the problem happens. In PSO algorithm, each particle swarm is a candidate solution to the problem. Such particles are distributed into a space of n dimensions (problem domain), each one having a determined position and velocity at each time instant. The best individual position of each particle is called personal best and the best position of all the particles is called global best. The performance of each particle is measured through a function called *rule function*, which varies according to the problem being dealt with. This function is the problem modeling in terms of PSO. It simulates the "environment" in which the individual is [17-18]. The learning of the particles is embedded in the position and velocity updating equations. At each time instant, the displacement of each particle is analyzed in search for the best position and its velocity is updated. The updating of these two factors depends on a comparison of the particle current position with its personal best and global best. The velocity equation also depends on random constants and the inertia weight. The inertia weight is related to how the velocity preserves the characteristics

of the previous instant, that is, how much it will influence the individual and collective knowledge, tending to a major or minor convergence [17-18]. The importance of inertia lies in the fact that the function has to avoid the early convergence in a personal best and provides the particles ways to find global best. This process occurs continuously until all the particles converge to the found global best, which is the best solution to the problem in question.

3.2 Definition of the Proposed PSO Algorithm

Each PSO particle represents a solution to a given problem. In this proposed approach, the problem is defined as the restoration of electric power distribution system, and the solution is defined as being the switches that must have their final status changed. Each particle is represented by a matrix, where the number of lines is equal to the total number of particles established and the number of columns is equal to the number of switches normally open. The switches may assume binary values, 0 or 1, where 0 means open switch and 1 means closed switch. It was verified through tests accomplished empirically with the computer program that 5 particles is the ideal number to work with. First, the numbers of particle colony is generated randomly, as well as the velocity and positions of the best locals and best globals. This is done by using random constants and considering the velocity null in the beginning. Then, at each iteration of the process, the velocities are updated and also the positions of the particles based on the velocities. The best locals and the best global are only updated in case they are considered better than the previous iteration. This is established by the rule function, which is equivalent to the modeling of the problem. This function measures the quality of each individual in the swarm. Due to the fact of working with discrete numbers, 0 or 1, the proposed velocity equation is normalized. At the end of each particle updating, if the position value is lower than 0.5, it is rounded to 0, otherwise it is rounded to 1. Nevertheless, with the purpose of increasing the number of velocity values possible and improving the program efficiency during its execution, the interval is extended to [-1;1]. This means that: closed switch is equal to 1 and open switch is equal to -1. This new interpretation may be simplified as: at the end of an updating of the particle position, if the position value is lower than 0.6, then it is rounded to -1, otherwise to, 1. The velocity equation is written so as to converge more slowly at the beginning of the iteration and more quickly at the end of the iteration. This guarantees the approximation of the particles to the best global and not to the local best. The formulas used in PSO are:

$$v(t+1) = ra * v(t) + (1 - ra) * [r * (l(t+1) - x(t+1)) + (1 - r) * ((g(t+1) - x(t+1))] \quad (1)$$

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

$$ra = (1 - t / n_it) \quad (3)$$

Where: i = counter for the number of particles; t = counter for the number of iterations; v(t) = particle velocity i in the iteration t; x(t) = particle position i in the iteration t; r = random constant, random number between 0 and 1 generated by the

program; l(t) = particle best local i found in the iteration t; g(t) = best global found in the iteration t; ra = inertia weight in the velocity equation; n_it = total number of iterations.

The rule function is constructed from the restoration problem in which an auxiliary vector is created. It assumes binary values 1 or 0, representing the status of each node, with or without energy, respectively. Each line of the system is represented by two consecutive nodes, according to Table 1. The function runs all the nodes to verify their status. The analysis is done from the status of the initial node and the status of its subsequent line. If the node is switched on (1) and its subsequent line is closed (1), then, this means that the line final node is also switched on. If the line is normally open, then the analysis will depend on the position of the particle related to it for present iteration. By applying this analysis to the whole system, then, all the switched on nodes are added. One of the criteria for algorithm optimization is to maximize the number of switched on nodes. Finally, the algorithm applies the criterion of minimization of the number of lines on. The best locals and the best globals update if the current position presents better performance in relation to the previous iteration.

4 Numerical Experiments and Methodology

The practical results associated to PSO are obtained in section. For this proposal, the large scale Distribution_System_01 is used [7], Fig. 1.

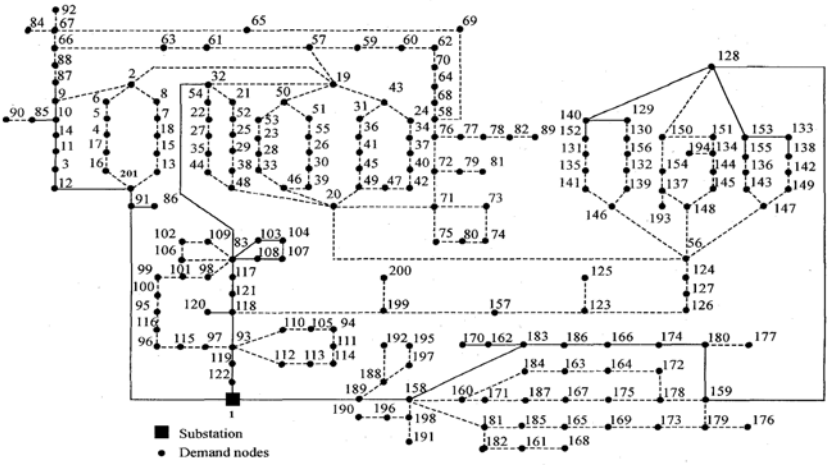


Fig. 1. Distribution_System_01with all the possible lines, including functional configuration information

As data input, the program has an archive which describes the system's energy lines in relation to the initial and final nodes that compose it. The status of the line is also described - 1 closed and 0 open. The archive is shown in Table 1.

Table 1. Program Input Data - description of the system's lines

Initial Node	*Final Node*	*Status*
1	91	1
1	122	1
1	189	1
2	6	1
2	8	1
2	9	0
2	19	0
3	11	1
3	12	1
4	5	1
4	17	1
5	6	1
7	8	1
7	18	1
9	10	1
9	87	1
10	14	1
10	85	1
11	14	1
12	201	1
13	15	1
13	201	1
15	18	0
16	17	1
16	201	1
19	32	1
19	43	0
19	50	1
19	57	0
20	46	1
20	48	1
20	49	1
20	56	0
20	71	1
21	32	1
21	52	1
22	27	1
22	54	1
23	28	1
23	53	1
24	34	1
24	43	0
25	29	1
25	52	1
26	30	1
26	55	1
27	35	1
28	33	1
29	38	1
30	39	0
31	36	1
31	43	1
32	54	1
32	83	1
33	46	0
34	37	1
35	44	0
36	41	1
37	40	1
38	48	1
39	46	1
40	42	1
41	45	1
42	47	1
44	48	1
45	49	1
47	49	1
58	68	1
58	69	1
58	76	0
59	60	1
60	62	1
61	63	1
62	70	1
63	66	1
64	68	1
64	70	1
65	67	1
65	69	1
66	67	1
66	88	1
67	84	1
67	92	1
71	72	1
71	73	1
71	75	1
72	76	1
72	79	1
73	74	0
74	80	1
75	80	1
76	77	1
77	78	1
78	82	1
79	81	1
82	89	1
83	98	0
83	103	1
83	106	1
83	108	1
83	109	1
83	117	1
85	90	1
86	91	1
87	88	1
91	201	1
93	97	1
93	110	1
93	112	1
93	118	1
93	119	1
94	105	1
94	111	1
95	100	1
95	116	1
96	115	1
96	116	1
97	115	1
98	101	1
99	100	1
99	101	1
102	106	1
102	109	0
103	104	1
104	107	0
105	110	1
107	108	1
111	114	1
112	113	0
113	114	1
117	121	1
118	120	1
118	121	1
118	199	1
128	159	1
129	130	1
129	140	1
130	156	1
131	135	1
131	152	1
132	139	1
132	156	1
133	138	1
133	153	1
134	144	0
134	151	1
134	194	1
135	141	1
136	143	1
136	155	1
137	148	1
137	154	1
137	193	1
138	142	1
139	146	0
140	152	1
141	146	1
142	149	0
143	147	1
144	145	1
145	148	1
147	149	1
150	151	1
150	154	1
153	155	1
157	199	1
158	160	1
158	181	1
158	183	1
158	189	1
158	198	1
159	178	1
159	179	1
159	180	1
160	171	1
160	184	1
161	168	1
161	182	1
162	170	1
162	183	1
163	164	1
163	184	1
164	172	0
165	169	1
165	185	1
166	174	1
166	186	1
167	175	0
167	187	1
169	173	1
171	187	1
172	178	1
173	179	0
174	180	1
175	178	1
176	179	1
177	180	1
181	182	1
181	185	1
183	186	1
188	189	1

Table 1. (*continued*)

50	51	1
50	53	1
51	55	1
56	124	1
56	146	0
56	147	0
56	148	1
57	59	1
57	61	0

119	122	1
123	125	1
123	126	1
123	157	1
124	127	1
126	127	0
128	140	1
128	150	1
128	153	1

188	192	1
188	197	1
190	196	1
191	198	1
195	197	1
196	198	1
199	200	1

Moreover, there is input of the system faulted line. The program aims to restore the system, providing an output line which must have its switch closed. It will also provide an analysis of the system with the fault situation before and following the reconfiguration. Three types of cases encountered are illustrated as follows.

Type 1 Case Study

This is the type of case analyzed in which the fault on the line causes a reduction in the number of loads supplied and there is a solution of closing the switch which enables an optimized restoration of the system. The program interface is divided into items described as follows:

- **Item 1:** Faulted line; this provides the program with data. Then, the program provides an analysis of the *fault situation,* showing the percentage of loads supplied and right below it shows the switched off nodes.
- **Item 2:** Program response; the program exhibits the execution time and the list with all the lines in the system after the solution – the initially switched on lines as well as the switched off lines and the one which had its status changed to restore the system.
- **Item 3:** Analysis of the solution found; the program exhibits the changed switch, followed by the number of loads supplied as well as the percentage of load supplied and the switched off nodes.

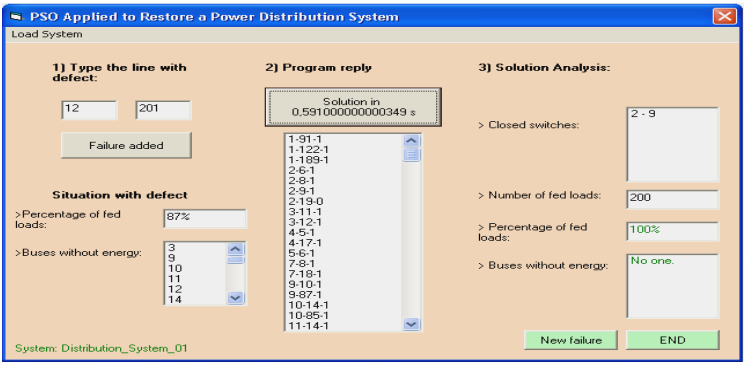


Fig. 2. Program response to fault presented on line 12-201 - Case Type 1

Fig. 2 shows the program interface for fault occurred on *line 12-201*- the analysis of the solution: number of loads supplied after the solution (maximum) and the num-

ber switched off nodes (zero). Table 2 presents several fault simulations on the system's lines for Type 1 cases and the program response to these faults.

Table 2. Fault Simulation on several lines -Type 1 Cases

Faulty Situation				Situation with the Program Solution				
Faulted Line		Nº of switched off Nodes	Loads Supplied%	Switch off		CPU Time [s]	Nº of switched off Node	Loads Supplied %
Initial Node	Final Node			Initial Node	Final Node			
1	91	40	80,00%	2	19	0,681	0	100,00%
1	122	87	56,50%	58	76	0,772	0	100,00%
1	189	73	63,50%	126	127	0,611	0	100,00%
9	10	19	90,50%	58	76	0,591	0	100,00%
12	201	26	87,00%	19	57	0,571	0	100,00%
16	17	8	96,00%	15	18	0,610	0	100,00%
19	32	10	95,00%	19	43	0,631	0	100,00%
19	50	9	95,50%	30	39	0,611	0	100,00%
20	71	13	93,50%	58	76	0,601	0	100,00%
21	32	35	82,50%	58	76	0,671	0	100,00%
25	29	32	84,00%	20	56	0,631	0	100,00%
32	54	4	98,00%	35	44	0,611	0	100,00%
32	83	50	75,00%	19	57	0,701	0	100,00%
42	47	5	97,50%	24	43	0,580	0	100,00%

Type 2 Case Study

This is the case analyzed in which the fault on the line causes the reduction in the number of loads supplied, but there is no possible solution to the problem, that is, there is no line to be closed which will maximize the loads supplied. No switch can be closed in order to maximize number of loads supplied. Thus, at the end of the solution the situation of the system remains the same as presented initially. Table 3 present fault simulations on the system's line for Type 2 cases and the program response to these faults.

Table 3. Faults on several lines Type 2 Case

Faulty Situation				Situation with the Program Solution			
Faulted Line		Nº of switched off Nodes	Loads Supplied %	Switch off	CPU Time[s]	Nº of switched off Nodes	Loads Supplied %
Initial Node	Final Node						
10	85	2	99.00%	There is no change	0.600	2	99.00%
67	84	1	99,50%	There is no change	0,601	1	99,50%
67	92	1	99,50%	There is no change	0,601	1	99,50%
72	79	2	99,00%	There is no change	0,601	2	99,00%
86	91	1	99,50%	There is no change	0,610	1	99,50%
118	120	1	99,50%	There is no change	0,601	1	99,50%
123	125	1	99,50%	There is no change	0,601	1	99,50%
137	193	1	99,50%	There is no change	0,601	1	99,50%
188	189	4	98,00%	There is no change	0,601	4	98,00%
199	200	1	99,50%	There is no change	0,580	1	99,50%

Type 3 Case Study

This is the case analyzed in which the fault on the line does not reduce the load supply, that is, the fault occurs on one of the normally open lines. Table 4 presents fault simulations on the system's lines for Type 3 cases and the program response to these faults.

Table 4. Fault on several lines of 3 Type Case

Faulty Situation				Situation with the Program Solution			
Faulted Line		N° of switched off Nodes	Loads Supplied %	Switch off	CPU Time [s]	N° of switched off Nodes	Loads Supplied %
Initial Node	Final Node						
15	18	0	100,00%	There is no change	0,581	0	100,00%
19	57	0	100,00%	There is no change	0,601	0	100,00%
20	56	0	100,00%	There is no change	0,591	0	100,00%
24	43	0	100,00%	There is no change	0,601	0	100,00%
56	146	0	100,00%	There is no change	0,581	0	100,00%
58	76	0	100,00%	There is no change	0,590	0	100,00%
112	113	0	100,00%	There is no change	0,581	0	100,00%
126	127	0	100,00%	There is no change	0,581	0	100,00%
164	172	0	100,00%	There is no change	0,591	0	100,00%
167	175	0	100,00%	There is no change	0,601	0	100,00%

5 Conclusion

This paper presents a novel strategy for the restoration of a distribution system through the use of an optimized reconfiguration using PSO. The proposed approach changes the logical status of a switch which is normally open. Moreover, it considers the maximization of number of loads supplied and the minimization of the number of switched on lines. The strategy disregards a total sweeping in the databank. Thus, less computational time is necessary to find the solution. This is essential for large scale systems, with greater possibilities of configurations. For this work, the number of cases candidates to the solution is 2 to the power of the number of lines normally open (26), resulting in 67108864 different configurations. The PSO algorithm proposed a velocity differentiated equation. The interval limit values were extended to -1 and 1. This was done with the purpose of augmenting the number of velocity values and improving the program efficiency. The most significant result of the algorithm is that it can be applied to any system topology. This makes the algorithm updating easier when an augment in system occurs.

Acknowledgments. The authors would like to thank CNPq, CAPES, and FAPEMIG - Brazilian research funding agencies, for the research scholarships, which supported this work.

References

1. Liu, Y., Gu, X.: Reconfiguration of Network Skeleton Based on Discrete Particle-Swarm Optimization for Black-Start Restoration. In: Proc. IEEE Power Engineering Society General Meeting, Montreal, Canadá (2006)
2. Coordinating Group of the Electrical Systems Planning: Decennial Plan of Expansion 1998/2007, Eletrobrás Report, Rio de Janeiro, Brazil (1998)
3. Rossi, R.: Systemic Hierarchic Classifier for Electrical Nets High-Voltage, Ph. D. Thesis, in Portuguese, Federal University of Itajuba, Itajuba, Brazil (2000)
4. Jardini, J.A.: Digital Systems for Automation of the Generation. In: Transmission and Distribution of Electric Energy, Portuguese. FCA Publisher, São Paulo (1996)
5. Experiences of CEMIG, ELETROPAULO, CESP, CPFL in Automation of the Distribution, I Conferência Internacional em Distribuição da Energia Elétrica, São Paulo, Brazil (1994)
6. Khushalani, S., Solanki, J.M., Schulz, N.N.: Optimized Restoration of Unbalanced Distribution Systems. IEEE Transactions Power Systems 22(2) (2006)
7. Ramírez-Rosado, I.J., Bernal-Agustín, J.L.: Genetic Algorithms Applied to the Design of Large Power Distribution Systems. IEEE Transactions Power Systems 13(2) (1998)
8. Esmin, A.A.A., Lambert-Torres, G., Zambroni de Souza, A.C.: A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization. IEEE Transactions Power Systems XX(Y) (2003)
9. Chin, H.-C. and Su, Y.-S.: Application of the Ant-Base Network for Power System Restoration. In: IEEE/PES Transmission and Distribution Conference & Exhibition, Dailian, China (2005)
10. Salazar, H., Gallego, R., Romero, R.: Artificial Neural Networks and Clustering Techniques Applied in the Reconfiguration of Distribution Systems. IEEE Transactions Power and Delivery 21(3), 1735–1742 (2007)
11. CIGRÉ: Pratical Use of Expert Systems in Planning and Operation of Power Systems, TF 38.06.03, Électra (146), pp. 30–67 (February 1993)
12. Buther, K.L., Momoh, J.A., Dias, L.G.: Expert System Assisted Identification of line faults on Delta-Delta Distribution Systems. In: Proc. 35th Midwest Simp. Circuits and Systems, vol. 2, pp. 1208–1213 (1992)
13. Hsiao, Y., Chien, C.: Enhancement of Restoration Service in Distribution Systems Using a Combination Fuzzy-GA Method. IEEE Transactions Power Systems 15 (2000)
14. Lambert-Torres, G., Quintana, V.H.: Intelligent System Application to Power System Problem Solving - Parts: I & II, Electrical Dept. & Computer Engineering, Waterloo University, Ontario, Canada (1996)
15. Bai, H., Zhao, B.: A Survey on Application of Swarm Intelligence Computation to Electric Power Systems. In: Proc. 6th World Congress on Intelligent Control and Automation, Dailian, China (2006)
16. Peng, X., Peng, Y., Dai, Y.F.: Swarm Intelligence Theory and Applications. Acta Electronica Sinca 31(12A), 1982–1988 (2003)
17. Kennedy, J., Eberhard, R.: The Particle Swarm: Social Adaptation of Knowledge. In: Proc. IEEE International Evolutionary Computation ICEC 1997, Indianapolis, USA, pp. 303–308 (1997)
18. Kennedy, J., Eberhard, R.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)

19. Nara, K., Kitagawa, M.: Distribution Systems Loss Minimum Reconfiguration by Simulated Annealing Method. In: Proc. IEEE Int. Conf. Advances Power Systems Control in Operation and Management, vol. 2, pp. 461–466 (1991)
20. Reynolds, C.W.: Flocks, herds and schools: A Distributed Behavioral Model. Computer Graphics 21(4) (1987)
21. Heppner, F., Grenander, U.: A Stochastic Nonlinear Model for Coordinated Bird Flocks; The Ubiquity of Chaos, AAAS, Washington (1990)
22. Hoffmeyer, J.: The Swarming body. In: Semiotics around the World; Proceedings of the Fifth Congress of the International Association for Semiotic Studies; Berkley (1994)

Space Regression Analysis on Geochemical Data by the GEP Evolutionary Model Based on Kriging

Dongmei Zhang[1,2], Ao Wang[1], and Zhifen Chen[1]

[1] School of Computer Science, China University of Geosciences, Wuhan, 430074, China
[2] State Key Laboratory of Geological Processes and Mineral Resources, China University of Geosciences, Wuhan, 430074, China
jjielee@163.com, wangao_1984@163.com

Abstract. There are several approaches to handle spatial trends: Kriging, stochastic simulation models, fractal and so on. The paper presents some contemporary approaches to spatial data analysis. The main topics are concentrated on the problems of space regression analysis by geochemical exploration data modeling. The innovative part of the paper presents integrated/hybrid model-combine GEP evolution modeling with spatial structure analysis. The models are based on GEP evolution modeling algorithm. Geostatistical tools on the basis of spatial autocorrelation thesis are used to extract representative data to fully utilize spatial structural information and weaken the influence of noise. Case study from mineral deposits in Gejiu illustrates the performance of the proposed model and BP neural network model is chosen as comparative study. It is shown that the fitting of the model and precision of test, provided by the combination of GEP evolution modeling and geostatistical model based approaches, are obviously improved.

Keywords: Spatial Distribution, GEP, Kriging, Spatial Autocorrelation.

1 Introduction

Spatial structure characteristics of geochemical exploration data are the results of various processes of geochemistry effects. The research of spatial structure will contribute to explore the reasons for further and improve understanding of geochemical processes. In the presence of trends, the data can be decomposed into two parts:

$$Z(x) = M(x) + e(x). \tag{1}$$

where $M(x)$ represents large-scale deterministic spatial variations(trends), and $e(x)$ represents small-scale spatial stochastic variations. Traditional approaches (e.g. Kriging, stochastic simulation models, fractal and so on) offer several possible ways to handle spatial trends and identify the structural characteristics of data spatial distribution, which imply a certain formula. The space distribution of the mineral deposit attribute information is influenced by a great deal of space factors, and there are high non-linear relations among the factors, so studying and introducing a new non-linear

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 239–247, 2008.
© Springer-Verlag Berlin Heidelberg 2008

method to analyze element spatial structure has important practical significance. Here, we will focus on an alternative way for trend modeling by using a data-driven approach.

The essence of space regression modeling is to describe the spatial evolution trend of information, according to a group of known real sampled point through the fitting of data to generate functional equations approximation the spatial distribution. The principle of evolution modeling is to construct a group of model expression at random according to the biological hereditary mechanism, and get the better model by a simple reproduction, crossover and mutation operator to adjust the structure of the model. Gene Expression Programming (GEP) is an amelioration method of genetic programming (GP), which has higher efficiency and accuracy than GP[1]. Because of existing ambiguity in mode features, fuzzy data, more noise and nonlinear characteristics of geochemical properties in spatial distribution model, we can use GEP, the principal advantages is the ability to discover patterns in complex data, to describe spatial distribution of geochemical data which exhibit significant unpredictable nonlinearity. In this paper, we propose a hybrid model(GEP evolution modeling with spatial structure analysis), which is first introduced and then extended for use in a combination with geostatistical models. The basic idea is unite a data-driven models which apply Kriging to extract representative data and use GEP to model large-scale nonlinear trends for mineralization forecast and reasonable evaluation. The remaining sections of this paper are organized as follows. In Section 2, we introduce GEP geochemical elements spatial structure regression modeling principle based on Kriging. The following section contains a brief description of GEP. Then, we present the case study of Yunnan Gejiu area Copper-tin metal deposits and comparison between BP, GEP, and Kriging-GEP(KGEP). The final section draws the conclusions and some directions for future work.

2 GEP Geochemical Elements Spatial Structure Regression Modeling Principle Based on Kriging

Traditional modeling methods divide sampling points into different modules for statistics. This kind of grid cell division method on the thought of taking random samples takes unit as a point, which ignores the geological structure, causing insufficiency of geological spatial structure of the information[2] . The implied spatial structure information of unit combination data is excavated randomly through GEP evolution modeling. Its geometric meaning is to resume a continuous ultra curved surface according to the designated sparse sample data, and adopt the mathematical surface to fit changing trend of data in the region and highlight local anomaly. But if only considering geographical coordinate x and y of the sampled point as input at the time of modeling, establishing non-linear functional relation according to corresponding geochemical attribute value as output, there is no complexity fully considering space distribution of information. In order to improve precision and dependability that

the mineral products predict, we must utilize spatial structural information related to mineral deposit.

Kriging method studies the distribution of variables which are characterized by randomicity and relativity in space using variogram function. It is determined that the around known data sets contribute to the unknown points through Kriging estimation. Semivariance of the difference between $Z(x)$ and $Z(x+h)$ which represent the value of regionalization variable $Z(x)$ at x and $x+h$(h means spatial distance) respectively acts as semivariance function $\gamma(x,h)$, under second-order stationary assumption:

$$E[Z(x+h)] = E[Z(x)], \forall h . \tag{2}$$

$$\gamma(x,h) = \frac{1}{2} E[Z(x) - Z(x+h)]^2 . \tag{3}$$

As the sample points are separated in practical application, formula (**3**) is rewritten as follow:

$$\gamma^*(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [z(x_i + h) - z(x_i)]^2 . \tag{4}$$

Where h represents step length, $N(h)$ represents the number of sample pairs with distance being h. Range, sill and nugget of the semivariance function will be determined by fitting experimental variogram functions with theoretical ones.

Since with Spatial autocorrelation thesis as theoretical foundation, drawing lessons from Kriging interpolation ideology, this paper puts forward GEP spatial regression modeling method (KGEP)which is based on inputting geographical coordinates and the known nearby sample points, considering the corresponding relationship between the points on the same coordinate and near points, through inserting spatial incomplete data to describe the spatial distribution trend of geochemistry attribute information[3]. Its function expression:

$$Z = f\left(x, y, A_1, A_2, \cdots, A_n\right). \tag{5}$$

Z expresses geochemical attribute values, x and y express geographical coordinates, A_1, A_2… A_n express the values of n known geochemical attribute nearest point. This method is similar to the characteristics of Kriging method. It can reflect the space-related among observed values offering a kind of definite estimation on the concrete spatial position in order to improve the simulation prediction accuracy of the model.

3 Gene Expression Programming Regression Modeling Design

GEP is a kind of new genetic algorithm on the basis of genome and phenotype group which was put forward by Portuguese scientist Candida Ferreira in 2001. GEP

genotype is fixed-length linear symbol made up by one or more genes; the phenotype means expression tree with different length and shape.

3.1 Gene Coding of GEP

Function sets {+, -, *, /, *Q*, *L*, *I*, *E*, *Q*, *S*, *C*, *T*, *s*, *c*, *t*}. +, -, *, / represent simple computing operations including addition, subtraction, multiplication and division, *L* expresses natural logarithm , *I* expresses 10^x , *E* expresses e^x , *Q* expresses evolution function, *S* expresses sine function, *C* expresses cosine function, *T* expresses tangent function, *s* expresses arcsine function, *c* expresses arcos function, *t* expresses arctan function. Finality symbol set include constants and variables in GEP programming. In Geological data modeling, you can choose the geological data in common use as a constant. Variables contain the attributes of all the samples such as $T=\{x_1,x_2,x_3,x_4,x_5,x_6\}$ .Definition the function set and terminator sets, by selecting elements from terminus set *T* and function set *F* to compose the head and selecting elements from terminus set *T* to compose rear for producing chromosome.

3.2 Fitness Function

In this paper, regression modeling uses square of the multiple correlation coefficient as a fitness function in the reference[4]:

$$R^2 = 1 - \frac{SS_E/(n-p-1)}{SS_T/(n-1)} . \tag{6}$$

Where *p* is independent variables; *n* is the number of data samples;

$\bar{y} = \sum y_i / n$, $\hat{y}_i = F(x_1, x_2, ..., x_n)$ is the fitted data of *i* dimension. Regression sum of squares: $SS_E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ and the total deviation sum of squares: $SS_T = \sum_{i=1}^{n} (y_i - \bar{y})^2$.

3.3 Algorithm Procedures of Gene Expression Programming

Step1. Create the initialization groups;
Step2. Calculate each individual's fitness function, and judge whether the function meet requirements of greatest fitness degree or number of iterations achieve target value. If it is consistent with the best individual output and the optimal solution, operation ends; otherwise turn to step3;
Step3. The following genetic manipulations will be executed with a new generation of stocks generated according to a certain probability, and return to step2;
1) Retain the best individual; 2) Selection; 3) Reproduction; 4) Mutation; 5) Transform and insertion sequence elements; 6) Recombination.

4 Case Study and Simulation

The space-based GEP regression modeling is studied through selecting Yunnan Gejiu area Copper-tin metal deposits as the research object, mainly GEP as modeling tool.

4.1 Experimental Steps

The basic idea is to use MORPAS etc. to preprocess and exploratory analysis, and use geostatistical tool to split data into training, testing and validation sets, then use GEP model to extract large-scale non-linear structures from data. In brief, the Kriging-GEP(KGEP) algorithm follows the steps given below:

(1) Study case from Yunnan Gejiu area Copper-tin metal mine area in-depth, and find out geochemical variables most closely related to mineralization.

(2) Divide the known mine areas and geological maps for forecasting into grid cells, and select variables from grid cells with MORPAS and MAPGIS.

(3) Use Kriging to extract representative data. Set a certain search radius and search the known points the nearest to all ones based on the known coordinates of training samples. Initialization of search radius and step length can accord to the sparseness degree of samples and the size of spatial samples. Considering a simple calculation, paper only selects the nearest eight points for calculation.

(4) GEP modeling which was used to model the trend in the data.

(5) Model test.

4.2 Enactment Parameters of Gene Expression Programming

In the present study, KGEP models with the following parameters were used:

(1) Set Generation with 10000 and PopSize with 50;

(2) Chromosome structure: the head length h equals to 11, the length of genes equals to 23, and each chromosome contains four genes. Using addition connecting subexpression tree;

(3) Genetic operators select probability settings: variation probability equals to 0.022, 1-point operator probability equals to 0.5, 2-point operator probability equals to 0.2, head-2-point operator equals to 0.1, IS transposition probability, RIS transposition probability and the gene transposition probability all equal to 0.1;

4.3 The Experimental Results and Analysis

(1) Spatial Regression of linear dimension

In this case, it was selected as training samples that 19 cell dataset of element Cu from mine points in study area. Firstly take GEP algorithm for modeling. The best adaptive value is 0.984221 in 10000 times' running, and the corresponding distillation function is:

T=(tg((2.292376*X_0)))*(cos(sqrt((((0.355002*X_0^3))*((3.374718*X_0)))-((0.245987*X_0^3)))))+(tg((2.218763*X_0^2)))*(((2.749802*X_0))+(sin(cos((0.245987*X_0^3)))))+(tg((1.657816*X_0^2)))*(sin((((0.245987*X_0^3))*((2.292376*X_0)))-

(((3.374718*X_0))+((1.443081*X_0)))))+exp(tg((cos(sin((3.266191*X_0^3))))+(cos(((0.568394*X_0))/((2.578771*X_0^3)))))+exp(tg(((sin((3.472258*X_0^2)))*(((3.846426*X_0))+((3.025697*X_0))))+(tg((0.036013*X_0)))))

X_0 means east-west coordinate x.

Then take KGEP algorithm on the same training samples. The best adaptive value is 0.991577 in the same time, and the corresponding distillation function is:

T=(sqrt((sqrt((1.181347*X_4^2)))+((3.015443*X_5^3))))+(((3.173778*X_4))+((2.181530*X_3^3)))+(0.133309*X_5)+((0.290301*X_2))+((sin((0.290301*X_2)))+((((1.228835*X_2))/(1.797107))-(2.795947)))+(tg((((0.688274*X_5^3))+((2.725508*X_8)))*(sin((0.790331*X_0)))))+(tg(((2.685955*X_4^2))*((0.954648*X_4^2))))+((tg((1.463590*X_3^2)))+(tg((2.685955*X_4^2))))+(tg(cos((0.195080*X_5))))

The parameters in the function represent east-west coordinate x and geochemical attribute values of the eight known points around the point separately. The mapping results (trend model) are presented respectively in Fig. 1 and Fig. 2.

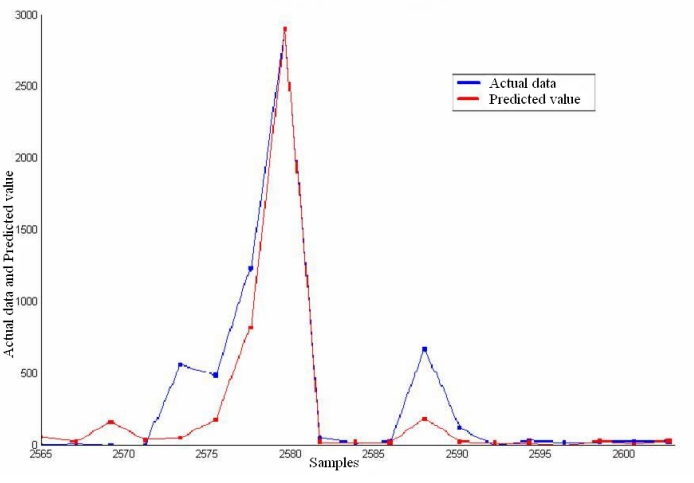


Fig. 1. Cu, GEP regression trend modeling

Compared Fig. 1 and Fig. 2, it can be found that KGEP regression trend modeling is better reflect changes in the overall space. Again, the performance of approaches (e.g. BP neural network, GEP algorithm and KGEP algorithm) on the same data sets is compared in terms of test results and relative errors(See Table 1).

As shown in the column of Table 1, the average error of BP algorithm valuation is 89.41%, the average error of GEP algorithm valuation is 57.58%, and it is only 36.87% for KGEP space regression modeling algorithm based on Kriging. KGEP algorithm provides the probability of success for this problem is 52.54%, 20.71% higher than BP and GEP algorithm. What's more, its model fitting and testing accuracy have been markedly improved. So the KGEP space regression modeling algorithm can reflect overall tendency of spatial in linear dimension.

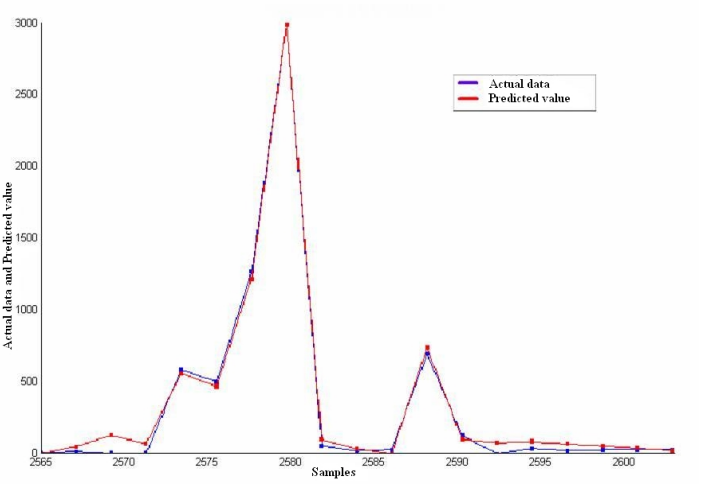


Fig. 2. Cu, KGEP regression trend modeling

Table 1. Samples Sets of Fitting Model and Fitting Results Comparison Table

Actual data			Predicted Value			Relative Error (%)		
sample	x	Ag	BP	GEP	KGEP	BP	GEP	KGEP
1	2565	62.2	196.8	38.6	38.2	216.42%	37.95%	38.12%
2	2567	70	197.7	193.6	90.7	182.42%	176.51%	29.66%
3	2569	57.9	200.6	49.7	167.5	246.46%	14.21%	189.35%
4	2571	59.3	210.4	137.3	109.9	254.85%	131.57%	85.35%
5	2573	633.3	246.2	476.7	605.8	61.13%	24.72%	4.33%
6	2575	550.7	407.3	459.3	513.7	26.03%	16.60%	6.71%
7	2577	1310	1376.9	1192.7	1278.3	5.11%	8.95%	2.42%
8	2579	3013.6	2898.7	3096.5	3070.5	3.81%	2.75%	1.89%
9	2581	111.3	111.3	10.4	137.7	0.00%	90.61%	23.76%
10	2583	74.3	84.3	145.2	74.7	13.46%	95.38%	0.48%
11	2585	84.7	126.4	2.9	39.1	49.25%	96.57%	53.88%
12	2587	744	151.2	809.3	792.9	79.68%	8.78%	6.57%
13	2589	180.9	160.6	4.3	138.6	11.23%	97.64%	23.38%
14	2591	55.4	163.7	19.2	117.2	195.44%	65.39%	111.57%
15	2593	91.2	164.6	21.8	122.5	80.52%	76.08%	34.29%
16	2595	78.3	164.9	90.1	108.1	110.64%	15.04%	38.06%
17	2597	82.3	165.0	42.1	95.0	100.52%	48.79%	15.47%
18	2599	85.2	165.1	68.7	77.7	93.73%	19.40%	8.84%
19	2601	84.6	165.1	191	61.3	95.11%	77.46%	27.57%

(2) Spatial Regression of two-dimension

In this case, it was selected as training samples that 14 cell dataset of element Ag from mine points in study area. Firstly take GEP algorithm for modeling. The best

adaptive value is 0.986269 in 10000 times' running, and the corresponding distillation function is:

T=(tg(sqrt(sqrt((1.182567*X_1^3)))))+(tg(cos(((2.910822*X_1^2))-((1.197095*X_1)))))
+((tg(sqrt((3.833242*X_1^2))))*(sin((2.121834*X_1))))*(sqrt(cos((3.778673*X_0^3))))+s
qrt((1.249466*X_0)+(tg(sqrt(sqrt((1.182567*X_1^3)))))+(tg(cos(((2.910822*X_1^2))-
((1.197095*X_1)))))+((tg(tg((1.376549*X_0^2))))+((2.362937*X_1^2)))+(sqrt(sin((1.1350
79*X_0^3))))

X_0 means east-west coordinate x, X_1 means north-south coordinate y.

Then take KGEP algorithm on the same training samples. The best adaptive value is 0.990078 in the same time, and the corresponding distillation function is:
T=(tg((3.918086*X_3^2)))+(((((2.432522*X_1^3))*((2.126961*X_9^2)))-
((2.986510*X_2^2)))+(cos(3.773057)))+((0.055423*X_3))*(cos(1.630471))+(0.179821*
X_5)+tg((sin((1.872795*X_0^3)))+((0.676921*X_2)))+(0.351706*X_1).

The parameters in the function represent east-west coordinate x, north-south coordinate y and geochemical attribute values of the eight known points around the point separately.

The performance of approaches (e.g. BP neural network, GEP algorithm and KGEP algorithm) on the same data sets is compared in terms of test results and relative errors (See Table 2).

Table 2. Samples Sets of Fitting Model and Fitting Results Comparison Table

Actual data				Predicted Value			Relative Error (%)		
sample	x	y	Ag	BP	GEP	KGEP	BP	GEP	KGEP
1	2579	18311	299.4	353.5	277.5	315.4	18.08%	7.31%	5.33%
2	2581	18285	107	95.2	87.1	93.2	11.04%	18.62%	12.92%
3	2583	18281	93.6	98.7	120.5	96.3	5.50%	28.77%	2.87%
4	2583	18283	142.4	98.1	119.9	110.8	31.11%	15.78%	22.15%
5	2585	18319	1298	1269.8	1305.3	1333.2	2.18%	0.56%	2.71%
6	2589	18307	54	88.6	71.6	66.4	64.09%	32.58%	23.00%
7	2591	18287	14.8	78.2	56.6	12.0	428.05%	282.61%	18.72%
8	2591	18299	29.7	48.6	111.4	27.9	63.82%	274.94%	6.03%
9	2595	18289	39.7	63.2	90.8	23.4	59.11%	128.80%	41.11%
10	2595	18309	101.3	183.8	51.3	111.0	81.47%	49.38%	9.62%
11	2597	18295	92.8	56.9	45.5	104.9	38.60%	50.99%	13.12%
12	2597	18305	15.1	54.5	144.7	10.5	260.66%	858.15%	30.48%
13	2599	18291	66.5	60.7	62.4	55.5	8.64%	6.21%	16.46%
14	2599	18297	53.8	54.6	60.3	53.8	1.45%	12.03%	0.03%

As shown in the column of Table 2, the average error of BP algorithm valuation is 76.39%, the average error of GEP algorithm valuation is 126.20%, and it is only 14.61% for KGEP space regression modeling algorithm based on Kriging. KGEP algorithm provides the probability of success for this problem is 61.78%, 111.59% higher than BP and GEP algorithm. So the KGEP algorithm can reflect overall tendency of spatial in two-dimension too.

5 Conclusion

Because of existing ambiguity in mode features, fuzzy data, more noise and other characteristics of geochemical exploration data in spatial distribution model, the actual observation data is inevitably interfered randomly, error, those make the relationship between the variables may not be accurately determined. Non-linear trends in geochemical data can be efficiently modeled by GEP. Combinations of GEP and geostatistical models gave rise to the decision-oriented risk and probabilistic mapping. Promising results presented are based on the case study: Yunnan Gejiu area Copper-tin metal deposits.. The hybrid model (GEP evolution modeling with spatial structure analysis) are preferable to pure GEP method because the former weakens the impact of noisy samples and considers spatial correlation between the observation values. The approach makes full use of local spatial structure information of the geological variables enhancing ability of describing the spatial distribution of attribute information. Experimental results prove that GEP space regression modeling based on Kriging is a new method with more reliable prediction on the spatial variables simulation, this method is an improvement and innovation of GEP algorithm and also a useful supplement to traditional Kriging valuation. Further, the algorithm needs to be improved if wanting to improve fitting accuracy of regression modeling and extrapolating ability of the model in practical application.

Acknowledgments. This paper was supported by the Research Foundation for Outstanding Young Teachers, China University of Geosciences (Wuhan) (CUGQNL0328) and the open issues of Geological and mineral resources of State Key Laboratory.

References

1. Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems 13(2), 87–129 (2001)
2. Zhang, Z., Xia, Q.: Modeling the Spatial Patterns of Mineralization Environments Using Fuzzy Sets. Earth science 30(1), 109–113 (2005)
3. Li, Q.: Study on Interpolation Method of Soil Spatial Information Based on RBF Neural Networks [master's paper]. Sichuan: Sichuan Agricultural University (2006)
4. Li, Q., Cai, Z.-h., Zhao, Y.: Application of Gene expression programming in prediction the Amount of Gas from Coal Face. Journal of basic science and engineering 2(1) (March 2004)
5. Jiang, S., Cai, Z., Li, Q.: A novel Parallel Gene Expression Programming algorithm Based on MPI. Acta Electronica Sinica 10(1) (October 2005)
6. Jia, X., Tang, C.J., Zuo, J.: Mining Frequent Function Set Based on Gene Expression Programming. Chinese Journal of computers 28(8), 1247–1254 (2005)
7. Chen, C.: Application and Research on mining Geological data based on evolution Calculation [doctoral dissertation]. Wuhan, China University of Geosciences (2006)
8. Kanevski, M., Parkin, R., Pozdnukhov, A., Timonin, V., Maignan, M., Demyanov, V., Canu, S.: Environmental data mining and modelling based on machine learning algorithms and geostatistics. Environmental Modelling & Software 19(9), 845–856 (2004)
9. Kanevski, M., Pozdnoukhov, A., Canu, S., Maignan, M.: Advanced Spatial Data Analysis and Modelling with Support Vector Machines. International Journal of Fuzzy Systems 4(1), 606–616 (2002)

Fault Tolerance Improvement through Architecture Change in Artificial Neural Networks

Fernando Morgado Dias[2,3] and Ana Antunes[1]

[1] Escola Superior de Tecnologia de Setúbal do Instituto Politécnico de Setúbal, Departamento de Engenharia Electrotécnica, Campus do IPS, Estefanilha, 2914-508 Setúbal, Portugal
Tel. +351 265 790000
[2] Departamento de Matemática e Engenharias, Universidade da Madeira Campus da Penteada, 9000-390 Funchal
Tel.: +351 291-705150/1; Fax: +351 291-7051993
[3] Centro de Ciências Matemáticas - CCM
Universidade da Madeira, Campus Universitário da Penteada
9000-390 Funchal, Madeira, Portugal
Tel.: + 351 291 705181; Fax: + 351 291 705189
aantunes@est.ips.pt, morgado@uma.pt

Abstract. This paper presents a technique for improving the fault tolerance capability of Artificial Neural Networks. This characteristic of distributed systems, which is usually pointed out as one of the advantages of this structure hasn't been deeply studied and can be improved in most of the networks. The solution implemented here consists of changing the architecture of feedforward artificial neural networks after the training stage while maintaining its output unchanged. It involves evaluating the elements of the Artificial Neural Network which are more sensible to a fault and duplicating inputs, bias, weights or neurons, according to the evaluation done before. This solution is very interesting because it allows maintaining the pre-trained network, but its cost is the need of additional hardware resources to implement the same network. The paper also presents an example of the application of the technique to illustrate its effectiveness.

1 Introduction

Fault tolerance (FT) is the capacity of a system to perform correctly under the presence of a fault. Graceful Degradation (GD) refers to the performance of a system that does not exhibit complete failure in the presence of a fault. This means that the performance might degrade due to a fault, but in a graceful way.

These two characteristics have been pointed out as advantages of the parallel processing of Artificial Neural Networks (ANNs), but whose real utility can only be evaluated through the use of specific hardware for ANNs.

For this evaluation it is necessary to define appropriate models for fault coverage. It might be possible to use the classical stuck at 0 and stuck at 1 or it might be useful to

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 248–257, 2008.
© Springer-Verlag Berlin Heidelberg 2008

define other types of models that are appropriate for ANNs. The discussion about how to model the faults in ANNs is still an open subject and took place, for example in [5], [7], [8] and [9]and recently re-gained actuality due to the possibility of using ANNs in critical nanosystems where FT and GD would be desirable characteristics [6].

Once models are chosen, an appropriate way of evaluating FT is selected and there is hardware available for the implementation, it is possible to study the solutions to improve the FT capabilities. At least the following possibilities exist:

- Change the training algorithms.
- Dynamic reconfiguration processes to be applied after a fault occurs.
- Modify the ANN after the training stage without changing its architecture.
- Modify the ANN after the training stage changing its architecture.

The possibility of changing the training algorithms to obtain a better FT was also the object of study of some researchers: [1], [2], [3].

The remaining solutions deal with the possibility of changing the ANN after the training stage. These can be implemented with or without changing the architecture of the network.

The dynamic reconfiguration option obliges the use of on-line training algorithms and also on-line detection of faults which is not at all a straightforward task to implement. This on-line detection of faults might be obtained through the use of an enlarged Boundary Scan scheme [4] and test information with extensive coverage to allow the identification of the fault and the re-organization of the ANN without the damaged elements.

The solution to improve after the training stage without changing the network's architecture is theoretically possible (at least in some nodes, but depending on the ANN structure), though difficult to implement since the changes must be such that the network's answers are the same before and after the change for every possible input.

The present work deals with the last possibility: modify the ANN after the training stage changing its architecture, as will be explained in the next section.

2 Improving Fault Tolerance through Architecture Change

The proposal for improving FT through the change of the ANN's architecture is based in a very simple algorithm:

1. Evaluate the importance of each element in the ANN to compute the output, according to a predefined criterium
2. Identify the parameter which is, by some predefined criterion, more relevant to the ANN output
3. Diminish the importance of this parameter without changing the ANN output in any circumstances.

This algorithm, however simple, raises several questions.

The first one concerns the evaluation of the importance of each element in the ANN to form the output. This requires a fault model to be used to simulate the faults and the proper comparison of the outputs in the presence and absence of a fault.

The discussion about the fault models has been done at different moments in the ANN community, see for example [5], [7]and [8], but here the fault model used will be the one from [9].

The second issue is how to rank the impacts in the output (especially if there are several situations to evaluate). Several possibilities could be used and the authors considered the worst case and the average error introduced for each element.

For the last point the issue is how to leave the output unchanged. The authors' proposal is made at the expense of hardware. Since a ANN is a distributed system and the final output is composed with the contribution of several units it should be possible to reduce the importance of each unit by splitting it into several other units of less individual importance to the final output.

3 The Fault Model Applied

Consider the ANN in figure 1. The inputs are noted as I_i, where I ranges from 1 to the number of inputs, N, the weights are w_{xy}, where x is the index associated with the inputs or the previous layer and y is the index associated with the neurons in the next layer.

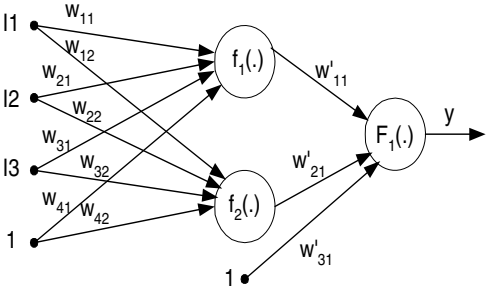


Fig. 1. Example of an ANN

The fault model used in the present work was proposed by [9]and is based in the stuck at models. The proposal of use for this model can be summarized as [9]:

1. For each connection test the effect of each individual fault by setting the weight associated with each connection to $+W_{MAX}$, $-W_{MAX}$ and 0.
2. For each input test the effect of a fault by setting the input to $+I_{MAXN}$, $-I_{MAXN}$ and 0.
3. For each neuron in the hidden layer(s) test the effect of a fault by setting all the weights deriving from the output of this neuron to $+W_{MAX}$, $-W_{MAX}$ and 0. This test is not needed in the case of MISO networks of only one hidden layer since then only one weight connects the output of each neuron.

Where W_{MAX} is the maximum value that the weights can assume with practical effect and I_{MAXN} is the maximum value that the input N can assume with practical effect.

4 The Algorithm Proposed

Once the evaluation of the importance of each element in the ANN to form the output and the identification of the parameter which is more relevant to the ANN output has been done, it is possible to increase the FT capabilities of the network.

Analysing the information of which element is responsible for the biggest impact in the output there are three possible situations:

- The error is in an input.
- The error is in a bias.
- The error is in a neuron.
- The error is in a single weight.

Depending on each situation a different action must be taken.

The general principle applied here is to diminish the importance of the fault. This results, according to the previous section, that some of the weights must be changed. The proposal is to split the weight or weights involved in two halves, setting the initial weight to $w_{xy}/2$ and include a new connection to supply the other half. With this operation the input of the activation functions will be the same as before so that no change is "visible" at the outputs.

The decision of splitting the weights involved in two halves is the one which requires less hardware. It would be possible to make a larger reduction at each step (for example splitting the weigh in three parts), using more resources but it could happen that this change would not change much in the overall performance. If another part of the ANN had almost the same FT, the overall improvement of splitting the weigh in three or more parts would be same as using two halves.

The Error in an Input

If the most important fault is located in an input, for instance input B, splitting the weight implies duplicating the input as shown in figure 2. In the resulting network each copy of input B has the weights deriving from it with half of the initial value.

This leaves the network producing the same output in all situations since the summation produces the same value as before, as can easily be observed from equation 1.

$$y = F(\sum_{i=1}^{n} Ii.wi) \qquad \text{Eq. 1}$$

The summation will have the same argument because of the contribution of both halves of the splitted weights.

It should be noticed that this change will also diminish the importance of all the other weights that are split, therefore increasing the overall FT of the ANN.

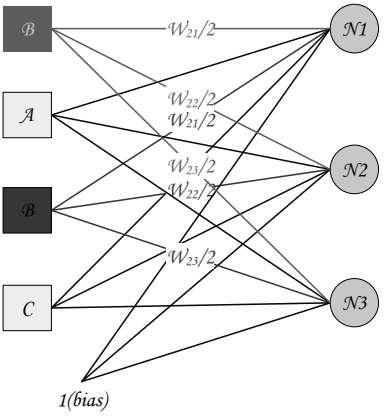


Fig. 2. Example of the ANN change in the case of a fault in the input

The Error in a Bias

If the most important fault is associated with a bias, the situation is very similar to the error in the input. It is possible to change only one weight and introduce another one to connect the additional input.

Alternatively it is possible to consider the bias as an additional input to all the neurons in that layer and duplicate all the weights deriving from this input with half of its original value. The later possibility is shown in figure 3.

Notice that this originates an unusual neuron with two or more bias, but this solution is effective to decrease the fault's impact in the output.

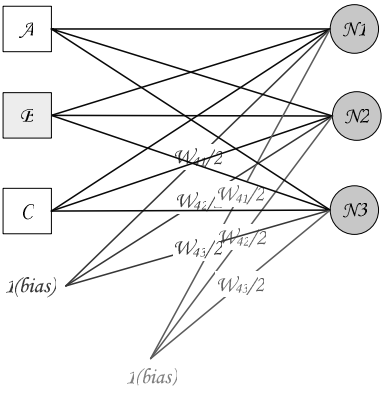


Fig. 3. Example of the ANN change in the case of a fault in a bias

The Error in a Neuron

If the error is in a neuron, all the weights deriving from this neuron will be affected. To solve the problem the neuron must be duplicated, including all of its connections to the previous layer and the connections coming from this neuron to the next layer

will have half their previous value, as can be seen from figure 4, where neuron 2 has been duplicated.

The Error in a Single Weight

If the error is in a single weight two situations must be considered: the weight is in the input layer or the weight is in the hidden layers.

In the first situation the procedure is similar to the one used for the error in the inputs (obliging to duplicate the input) and the second situation is similar to the error in the neuron (obliging to duplicate the neuron).

It should be noted that each change in ANN affects several weights and therefore, for each iteration, the impact of the faults in the output must be recalculated.

As an example of the cross effects of the changes, notice that from figure 4 it can be seen that an error in weight W_{B1} or W_{B2} would have resulted in the same changes and both the weights' impact in the output is reduced as a result of this change.

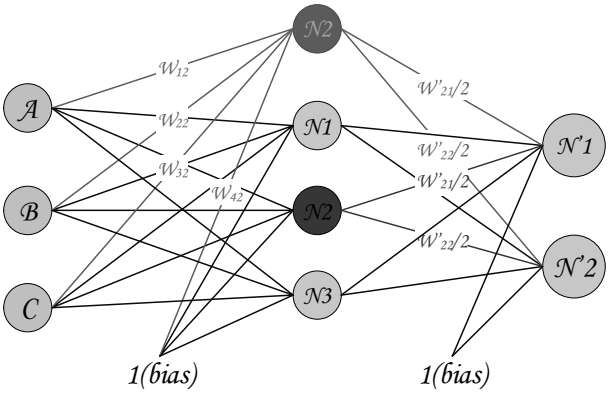


Fig. 4. Example of the ANN change in the case of a fault in a neuron

5 Practical Examples

As an example of the application of this technique of FT improvement, consider the following network:

- 3 inputs.
- 1 hidden layer with 2 neurons.
- 1 output neuron.
- The hidden layer's activation functions are hyperbolic tangents.
- The output neuron's activation function is a linear function.

For the evaluation of the FT, the range of the inputs is needed. The range of both inputs can be seen in the matrices 1.

$$[-5 \quad 5]\ ;\ [-3 \quad 7] \qquad \text{mat. 1}$$

The weights associated with each layer of the ANN can be seen in the matrix 2.

$$\begin{bmatrix} 0.5 & -0.9 \\ -0.1 & 0.3 \\ 0.9 & -0.4 \end{bmatrix} [0.5 \quad -0.7 \quad 0.8] \qquad \text{mat. 2}$$

The structure of the ANN is shown in figure 5.

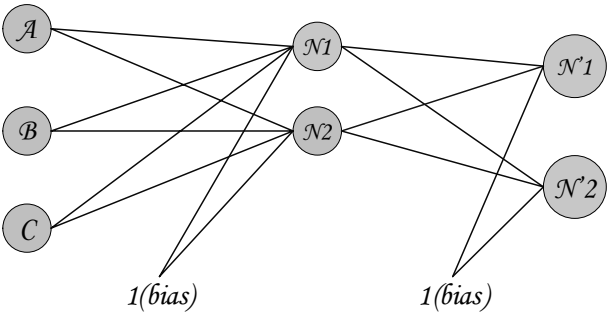


Fig. 5. Initial network of the example

The evaluation of the FT is done according with equation 1, which gives the relative error for each fault.

$$Error = \frac{output_with_fault - output_without_fault}{output_without_fault} \qquad \text{Eq. 2}$$

For this ANN there are 9 combinations of the inputs and it is necessary to evaluate the correct outputs for these combinations.

The highest peak error (one of the possibilities mentioned in section 2) obtained for this network is 6.97. To obtain this value it is necessary to calculate every single error mentioned in section 3. That means testing every type of error and setting the weights involved to $+W_{MAX}$, $-W_{MAX}$ and 0 for the weights and to $+I_{MAXN}$, $-I_{MAXN}$ and 0 for the inputs.

The most sensible parameter for this network is the weight connecting the bias input to the output neuron. This bias can be duplicated according to what was proposed in section 4.

After the duplication of this input, the highest peak error is now reduced to 6.1363, which does an improvement of 12% relatively to the original network.

Another step of this algorithm reveals that the most critical point is now the output of neuron 2 at the hidden layer. The procedure now implies that this neuron must be duplicated.

After introducing the new neuron and resizing the weights the new FT evaluation reveals that the highest peak error is now of 5.33, resulting in an improvement of 13%.

The global improvement for the two steps rises to 24%. Naturally the improvement of tolerance came at the cost of using more hardware than it was used at the initial implementation.

Figure 6 shows the final structure of the ANN of the example, after two iterations of the proposed algorithm, with an additional neuron and an additional bias input.

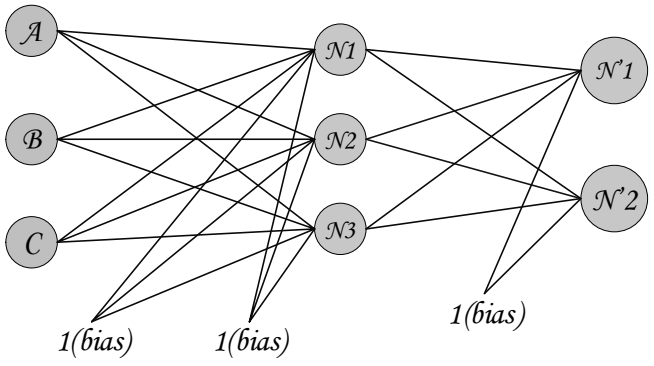


Fig. 6. Structure of the network after two steps of improvement

Another Example

As a second example, consider the following network:

- 3 inputs.
- 1 hidden layer with 4 neurons.
- 1 output neuron.
- The hidden layer's activation functions are hyperbolic tangents.
- The output neuron's activation function is a linear function.

For the evaluation of the FT, the range of the inputs is needed. The range of both inputs can be seen in the matrices 3.

$$[-3 \quad -1][-5 \quad 2][-10 \quad -1] \qquad \text{mat. 3}$$

The weights associated with each layer of the ANN can be seen in the matrix 4.

$$\begin{bmatrix} -0.5 & 0.3 & 1.1 & 0.2 \\ -1 & -0.1 & -0.1 & 0.5 \\ 1.3 & -0.8 & -0.7 & -0.6 \\ -0.4 & 0.3 & 0.2 & -0.9 \end{bmatrix} \quad [0.3 \quad -0.2 \quad 0.1 \quad 0.2 \quad 0.3] \qquad \text{mat. 4}$$

The highest peak error obtained for this network is 143.32. The parameter responsible for this error is the weight connecting input 3 to neuron 3.

After the duplication of this input, the highest peak error is now reduced to 90.22, which does an improvement of 37% relatively to the original network.

Another step of this algorithm reveals that the most critical point is now the output of neuron 2 at the hidden layer.

After introducing the new neuron and resizing the weights the FT evaluation reveals that the highest peak error is now of 77.09, resulting in an improvement of 14%.

The global improvement for the two steps rises to 46%.

6 Discussion

It has been pointed out that the algorithm proposed here produces an increase in FT using additional hardware. In some situations the use of additional hardware might not be associated with an increase in the cost of the implementation. The hardware implementations can be done with commercial solutions for ANNs (for an overview of options see [10]) or a specific implementation. In both cases, since the hardware is not tailored to any specific size, it is very likely that some resources are left unused and they can be used to improve FT without any additional cost. An additional input or one more neuron probably still fit in the piece of hardware and can be used to improve the FT.

It would be interesting to study the improvement of FT versus the associated cost with this improvement but this study cannot be carried out without a specific application in view. Only with such application it would be possible to evaluate the cost of using hardware with more resources because it would be dependent of the type of implementation (Application Specific Integrated Circuit (ASIC), Field Programmable Gate Array (FPGA), commercial solution, etc), architecture, precision and several other factors.

7 Conclusion and Future Work

Fault tolerance is a very important characteristic of ANNs. In critic systems it is not a feature, it is mandatory.

The present work introduces a technique for improving the FT capabilities of an ANN at the expense of hardware. It is based in duplicating parts of the network to decrease the impact that a certain weight has in the ANN's output. The cost of this technique is only the additional hardware that is needed to implement the same ANN. The ANNs are left unchanged regarding their output and therefore no retraining is needed. The technique is based in lowering the importance of the parts of the ANN that are more significant to compute the output. After the analysis of these parts and which elements they affect, the corresponding weights are split in two halves.

The application of the technique to two very simple example networks is introduced to show its utility. The technique is applied in two steps which improve the FT by 24% and 46%, but requires additional hardware.

The study of the cost of the additional hardware can only be done in the presence of a specific solution. It will depend on the additional resources needed and on the type of implementation used: ASIC, FPGA, commercial solution, etc.

As future work the authors will prepare a software tool that implements this technique of improving the FT capabilities of an ANN so that the ANN community can

test and make use of it. This software will be able to take an ANN architecture and weights and return the parameters and architecture of the improved network.

References

1. Cavalieri, S., Mirabella, O.: A novel learning algorithm which improves the partial fault tolerance of multilayer neural networks. Neural Networks 12, 91–106 (1999)
2. Elsimary, H., Mashali, S., Shaheen, S.: Performance evaluation of a novel fault tolerance training algorithm. In: Proceedings of the World congress on computational intelligence, vol. 2, pp. 856–860 (1994)
3. Elsimary, H., Mashali, S., Shaheen, S.: Generalization ability of fault tolerant feed forward neural nets. In: Proceedings of the IEEE International conference on systems, man and cybernetics, vol. 1, pp. 30–34 (1995)
4. Parker, K.: The Boundary Scan Handbook (1992)
5. Bolt, G.: Investigating the Fault Tolerance in Artificial Neural Networks., Technical Report YCS 154 from the University of York, available at the internet (1991)
6. Eickhoff, R., Rückert, U.: Tolerance of Radial Basis Functions Against Stuck-at-Faults. In: International Conference of Artificial Neural Networks, Poland, pp. 1003–1008 (2005)
7. Phatak, D.S.: Complete and Partial Fault Tolerance of Feedforward Neural Nets. IEEE Transactions on Neural Networks 6/2, 446–456 (1995)
8. Tchernev, E.B., Phatak, D.S.: Investigating the Fault Tolerance of Neural Networks, NCA (2005)
9. Dias, F.M., Antunes, A.: Fault tolerance of feedforward neural networks: an open discussion for a global Model. Engineering Applications of Artificial Intelligence (submitted, 2007)
10. Dias, F.M., Antunes, A., Mota, A.: Artificial Neural Networks: a Review of Commercial Hardware. Engineering Applications of Artificial Intelligence, IFAC 17/8, 945–952 (2004)

A Strategic Model for Measuring Agility with Fuzzy Logic

Gholamreza Khoshsima

Instructor, Department of Industrial Management, Vali-E-Asr University, Rafsanjan, Kerman, Iran
Khoshsima@vru.ac.ir

Abstract. Agility is the set of capabilities and competencies that an organization needs to thrive and prosper in a continuously changing and unpredictable business environment. This paper, based upon our studies in Iran's Manufacturing Industry, presents a model for organizational agility and then suggests a knowledge-based methodology for measuring agility. For being agile, every organization needs a number of capabilities (e.g. responsiveness) and competencies (e.g., knowledge management (KM) and flexibility), these capabilities and competencies are broken into some dimensions and finally the resulting dimensions to some indices. For measuring agility we use "IF {Fuzzy antecedents} THEN {Fuzzy consequent}" rules. This system uses expert knowledge and includes implementation of fuzzy logic methods and terminology.

Keywords: Agility, Measurement, Fuzzy Logic.

1 Introduction

The nature of competition has changed dramatically over the last two decades. It is often stated that the most critical concept for a manufacturing company today is its capability and competency for achieving sustainable competitive advantage. In the present turbulent, continuously changing and unpredictable business environment, agility is increasingly becoming the greatest tool of organization. How can organizations survive and acquire maximum advantage in this environment? The answer is agility; agility is one of the most important concepts for the organizations to thrive in a turbulent, continuously changing and unpredictable environment. Agile in Webster means "marked by ready ability to move with quick easy grace" and "having a quick resourceful and adaptable character" and Agility means "the quality or state of being agile: Nimbleness, Dexterity"[16].

Traditionally, performance measurement is defined as the process of quantifying effectiveness and efficiency of action into readable symbols to report. The need to measure agility has taken on an increasing importance as effective comparison of performance can only be carried out when appropriate measures have been designed and tested [36]. In modern business management, performance measurement goes well beyond merely quantification and accounting. It supposed to contribute much more to business management and agility improvement in the industries [4]. Form the

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 258–268, 2008.
© Springer-Verlag Berlin Heidelberg 2008

management perspective, performance measurement provides the necessary information for management feedback for decision makers and process managers. It plays a critical role in monitoring performance, enhancing motivation and communication, and diagnosing problems [4][54]. Furthermore, performance measurement provides an approach to identifying the success and potential of management strategies, and facilitating the understanding of the situation. It assists in directing management attention, revising company goals, and re-engineering business processes [4][52][3][25]. Performance can be defined as the ability of an object to produce results in a dimension determined a priori, in relation to a target. Thus it is necessary to have, first, an object whose performance is to considered; second, a dimension in which one is interested; and third, a set target for the result. The presence of these three elements ensures that performance as defined above does exist [26]. The journey to agility is a never-ending quest to do better than the competition, even as the competitive environment constantly changes. On this journey one never arrives at a final destination, but one needs to know at any point in time whether progress is in fact in the right direction, and how one stands relative to worldwide competition [13].

Fuzzy logic provides a useful tool to deal with problems in which the attributes and phenomena are imprecise and vague [56]. Furthermore, fuzzy logic has found large application in management decisions. Manufacturing systems engineering lacks analytic and closed-form mathematical solutions albeit in the simplest possible cases. Since manufacturing systems are operated and managed by people, it is necessary to record and utilize human knowledge and perceptions about agility and its factors (parameter quantification and measurement). However, due to the imprecise and vague definition of agility indicators, and when a situation is characterized by either lack of evidence or the inability of the experts to make a significant assessment of an event, linguistic expressions are used to estimate ambiguous events. Most agility measurements are described subjectively by linguistic terms, which are characterized by ambiguity and multi-possibility. Thus, the scoring of the above techniques can always be criticized, because the scale used to score the agility capabilities has two limitations: 1) such techniques do not take into account the ambiguity and multi-possibility associated with the mapping of one's judgment to a number, and 2) the subjective judgment and the selection and preference of evaluators have a significant influence on those methods. However, fuzzy logic provides a useful tool to deal with problems in which the phenomena are imprecise and vague. Using fuzzy concepts, evaluators can use linguistic terms to assess the indicators in a natural language expression and each linguistic term can be associated with a membership function. Fuzzy logic by making no global assumption about the independence, exhaustiveness, or exclusiveness of facilitated evidence, tolerates a blurred boundary in definitions. This brings hope of incorporating qualitative indicators into decision-making since it is often vaguely defined or has unclear boundaries. Furthermore, fuzzy logic has found large application in management decisions. From this review, to assist managers in better achieving an agile enterprise, a model on the basis of fuzzy logic is purposed to provide a means of both measuring how agile an enterprise is and identifying the principal obstacles to improve the agility level [30]. Algebraic formulae fail in putting together the various dimensions of agility coupled with the human perception of agility. To overcome such problems, the key idea is to model human inference, or equivalently, to imitate the mental procedure through which experts (managers, engineers, operators, and

researchers) arrive at a value of agility by reasoning from various sources of evidence. To quantify agility, managers and operators, frequently use verbal or linguistic values, such as low, average, about high and so on. Thus, a valid and suitable candidate solution to the problem of measuring enterprise agility should be based on fuzzy logic [49][48][47]. This approach has several advantages when compared to previous methods: 1) this method can give the analyst relatively realistic and informative information. This provides an overall picture about the possible agility of an organization and ensures that the decision made in selection will not be biased. 2) this method can systematically identify the weak factors within an organization and provide the means for a manager to formulate a comprehensive plan for improvement. Therefore, the method can be further used in self-assessment. 3) It provides an appropriate function structure to reflect the imprecise phenomena in many business environments and takes into account the uncertainty effect of each factor to ensure a more convincing and reliable evaluation. Moreover, algorithm of the proposed method can be computerized. Thus, by the decision makers' providing linguistic assessments through a menu-driven interface design, the agility level of an enterprise and its agility obstacles can be obtained easily [30]. The purpose of the study is to assess a firm's agility using a reasonably comprehensive model with fuzzy logic.

2 Competencies and Capabilities

Selznick and Penrose, the first time, introduced the notion of competence in 1957 and 1959[31][35][15][39]. Prahalad and Hamel presented basic concepts of competency and capability in 1990. Core competency is collective learning in the organization, especially how to coordinate diverse production skills and integrate multiple streams of technologies and is communication, involvement, and a deep commitment to working across organizational boundaries. It involves many levels of people and all functions. In fact, the corporation, like a tree, grows from its roots. Core products are nourished by competencies and engender business units, whose fruit are end products[40]. Some practitioners differentiate between competencies and capabilities [31][51][44][19][8][53] [15] and some of them do not. Competency[40][31][44][19] we mean to include what others have called "component competence"[15], resources[15][1], and "knowledge and skills" and "technical systems"[15][45][28], while by using the term capability [31][51][50][44][8][1], we mean to include what others have called integrative capabilities[15][27], dynamic capabilities[15][46][45][12],"implicit/social" or "collective knowledge"[15][43], "organizational architecture"[15][37], combinative capabilities[15] [21], "managerial systems" and "values and norms"[15][28], invisible assets[15][18], and architectural competence[15]. In this paper, we too differentiate between them. In our approach, organizational capability is a firm's ability to establish internal structures and processes that create firm-specific competencies and enable it to adapt to changing external pressures. Organizational capability focuses on the internal environment of the firm, on decisions and practices within the firm that modify how it responds to external pressures[51]. Capabilities are thus associated with groups of individual competencies that collectively turn into organizational competence. According to Dobson & Starkey (1993) the capability of any organization is fundamentally determined by how well it

links together its various activities-such as designing, marketing, and supporting its products and services and also it depends on the quality of coordination across these activities, not just on competence in each individual activity[8]. Stalk et al in 1992 reviewed competencies and capabilities[44]: *Competencies and Capabilities represent two different but complementary dimensions of an emerging paradigm for corporate strategy. Both concepts emphasize behavioral aspect of strategy. However, core competencies emphasize technological and production expertise at specific points along the value chain, but capabilities is more broadly based, encompassing the entire value chain. In this respect, capabilities are visible to the customer in a way that core competencies rarely are.* Therefore capabilities/competencies are collective learning especially includes of set of abilities, expertise, knowledge and skills at organization. These capabilities and competencies create competitive advantages for organizations. If customer could see its, therefore its capabilities and if couldn't see it's for example roots of tree, its competencies.

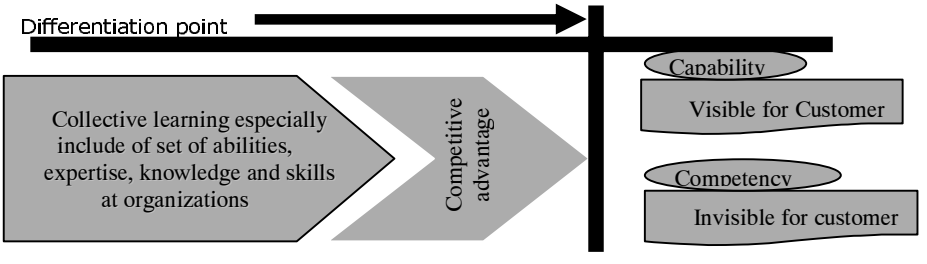


Fig. 1. Concepts of capabilities and Competencies

3 Conceptual Model

Agility is a set of capabilities and competencies that an organization needs to thrive and prosper in a continuously changing and unpredictable business environment. Based on this definition, two capabilities (Flexibility and Responsiveness) and a competency (KM) are vital to be agile. Therefore, agile organization includes three components: flexibility, responsiveness and KM. In other words, if an organization is responsive, flexible and KM is appropriately implemented, this organization is agile (Fig 2). Responsiveness is visible to customers but flexibility and knowledge management are not. The model covers four dimensions of Goldman et al model[13]: Enriching customers, Cooperating to enhance competitiveness, Organizing to master change and uncertainty, and Leveraging the impact of people and information. Sharifi and Zhang[41][42] presented four agility capabilities (Responsiveness, Flexibility, Competency and Speed) and Dove in 1999 has introduced two competencies (KM and CP). Nevertheless, they have not differentiated between competencies and capabilities in their perspectives. One of this paper's objectives is to redefine the following operational concepts, which are already offered as key words: **Responsiveness** is a significant component of agility[24][33][51] [41][42]. Responsiveness is the ability of presenting product/services rapidly to the customer. **Flexibility** also is one of the most important components of agility[41][42][34]. Flexibility is ability of coping and

adapting with the environment. **KM** is another important component of agility[9][55]. KM is responsible for having the right knowledge in the right place at the right time[9]. For measuring each of dimensions used four components of people, processes, infrastructure and strategies.

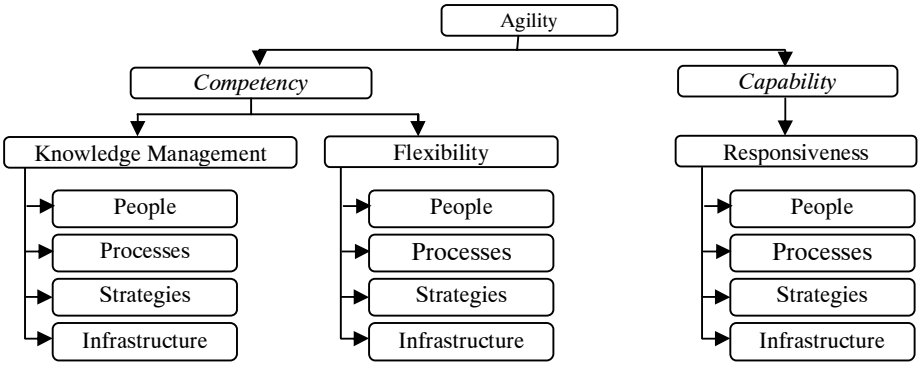


Fig. 2. Conceptual model for achieving agility

4 Fuzzy Approach

Fuzzy logic simplifies the process of taking decisions by simulating the way of reasoning of a human expert in environments characterized by uncertainty and imprecision. The idea behind of fuzzy logic is that an element can belong partially to several subsets, unlike Boolean logic where belonging or not to a set are mutually exclusive. The degree of belonging to a set is a value between 0 and 1, usually determined by to what extent an element belong to a fuzzy subset or a category of a variable. Fuzzy logic is applied in the building of fuzzy systems, which establish the relationship between an input space and an output one. The breakthrough regarding traditional mathematical models lies in the fact that the relationship is not determined by complex mathematical equations, but by means of a set of logical rules that reflect the way of reasoning of an expert. These rules consist of an antecedent (in which several input variables are related by means of logical operators) and a consequent (where the same process occurs amongst the output ones). Once defined the membership functions and the rules, the fuzzy inference process occurs in several steps, as follows (further background in fuzzy logic can be found for example at [11][6][2][10]): 1) Fuzzification: The first step is to identify for each value of the input variables, the degree of membership registered in each established label or category. 2) Aggregation of antecedents: Once known the values registered in the labels present in the antecedents, different methods of aggregation can be used in order to obtain a unique global degree of truth for the antecedent. 3) Inferencing: From the global degree of truth of the antecedent, a membership function can be derived based on the membership function of the label of the output variable present in the consequent. For achieving this,

several methods can be applied, the most accepted being PROD (which weights the membership function in the consequent by the value of the degree of truth of the antecedent) and MIN (which truncates the function of the activated label in the consequent according to the value of the degree of truth of the antecedent). 4) Composition: As several rules may affect the same output variable, it is necessary to look for a way of aggregating the membership functions obtained in the inference of all the rules. The most common methods are SUM (which offers as the final membership function the sum of the ones obtained after inferencing all the rules), MAX (which offers a function that takes in each point of the output domain, the maximum value of the ones obtained in each particular membership function) and PROBOR (which is very similar to SUM, but offers the sum of the values obtained in each output variable minus the end result of their multiplication). 5) Defuzzyfication: Finally, a method has to be applied for converting the membership function obtained in the previous step into a crisp value. Some common methods consist in taking the output value corresponding to the minimum, medium and maximum of the maximums of the membership function. The bisectrix method, however, looks for the value that divides the surface between the output function and the x-axis into two sections with the same area. However, the most adopted method is the centroid, which offers as the output value the x coordinate of the center of gravity of the surface between the function and the x axis [14].

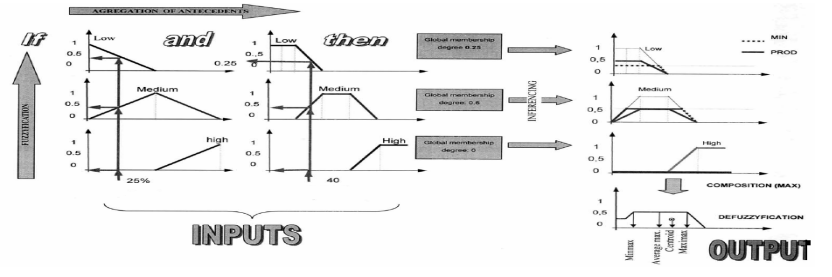


Fig. 3. Scheme of fuzzy inference process[14]

Five categories of impact were defined for the input/output variables. With respect of questions in somewhere were used from ‘very low’, ‘low’, ‘medium’, ‘high’ and ‘very high’ (shown at Fig 4) and in another somewhere were used form ‘very poor’, ‘poor’, ‘fair’, ‘good’ and ‘very good’. The boundaries between these categories (once again considering trapezoidal functions and the sum of degrees of membership equal to the unit) were established as shown in Fig. 3. The respondents were asked to answer the questionnaire using a 5-point Likert scale. Hwang and Lai [17], Chen and Hwang [5] and Liang and Wang [29] turned Likert scale into fuzzy number. We used the Liang and Wang’s number. The membership functions of the linguistic values are as follows:

$$\mu_{VL}(x)=\begin{cases}1-\dfrac{10x}{3} & 0\le x\le 0.3\\ 0 & otherwise\end{cases}$$

$$\mu_{L}(x)=\begin{cases}\dfrac{10x}{3} & 0\le x\le 0.3\\ \dfrac{5}{2}-\dfrac{10x}{3} & 0.3\le x\le 0.5\\ 0 & otherwise\end{cases}$$

$$\mu_{M}(x)=\begin{cases}\dfrac{10x}{3}-\dfrac{2}{3} & 0.2\le x\le 0.5\\ \dfrac{8}{3}-\dfrac{10x}{3} & 0.5\le x\le 0.8\\ 0 & otherwise\end{cases}$$

$$\mu_{H}(x)=\begin{cases}5x-\dfrac{5}{2} & 0.5\le x\le 0.7\\ \dfrac{10}{3}-\dfrac{10x}{3} & 0.7\le x\le 1\\ 0 & otherwise\end{cases}$$

$$\mu_{VH}(x)=\begin{cases}\dfrac{10x}{3}-\dfrac{7}{3} & 0.7\le x\le 1\\ 0 & otherwise\end{cases}$$

$$\mu_{VP}(x)=\begin{cases}1-5x & 0\le x\le 0.2\\ 0 & otherwise\end{cases}$$

$$\mu_{P}(x)=\begin{cases}5x & 0\le x\le 0.2\\ \dfrac{7}{2}-5x & 0.2\le x\le 0.4\\ 0 & otherwise\end{cases}$$

$$\mu_{B}(x)=\begin{cases}5x-\dfrac{3}{2} & 0.3\le x\le 0.5\\ \dfrac{7}{2}-5x & 0.5\le x\le 0.7\\ 0 & otherwise\end{cases}$$

$$\mu_{G}(x)=\begin{cases}5x-3 & 0.6\le x\le 0.8\\ 5-5x & 0.8\le x\le 1\\ 0 & otherwise\end{cases}$$

$$\mu_{VT}(x)=\begin{cases}5x-4 & 0.8\le x\le 1\\ 0 & otherwise\end{cases}$$

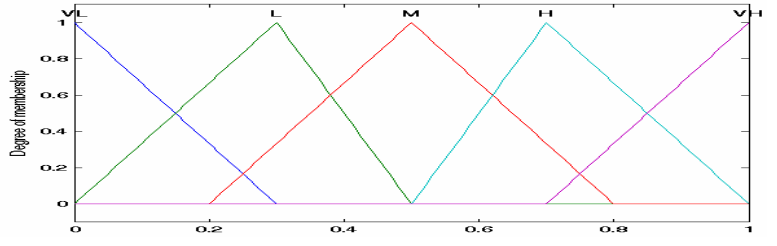


Fig. 4. Membership function

The core of the fuzzy inference system is based on the set of rules defined to link input and output spaces, in an attempt to simulating the way an expert reasons[10]. In this case, all the rules have the same structure. The antecedents link all the possible combinations of categories in qualitative and quantitative variables by means of the operator AND. These combinations were linked to the expected value of the output variable (the corresponding impact) according to the criterion established, although the set of rules was finally revised by the panel of experts to ensure the validity or to modify the standard criterion for exceptional cases. There are different methods of defuzzification, the final step of the fuzzy inference process. The most widely accepted (and thus the ones that we have chosen) are minimum of maximums, medium of maximums, maximum of maximums and centroid [14].

The key idea of our model in measurement is the involvement of all distinct types and corresponding operational parameters in the determination of the overall agility. This is implemented via multi-antecedent fuzzy If-Then rules, which are conditional statements that relate the observations concerning the allocated types (If-part) to the value of agility (Then-part). Suppose A_i, i = 1, 2…N, is the set of agility components and LC_i the linguistic value of each component; then the expert knowledge general rule is

If A_1 is LC_1 AND … A_N is LC_N Then A_{ORGA} is $ORGA$.
Where $ORGA$ representing the set of linguistic values for organizational agility A_{ORGA}. All linguistic values A_i and $ORGA$ are fuzzy sets like what is shown in Figs 3 and 4.

As can be seen from Figure 2, agility breaks into three components/dimensions (e.g. flexibility, responsiveness and knowledge management into). In addition, each dimension breaks into four components include of infrastructure, strategy, people, and process and each of this components measuring with measurable indices. At the first and second levels, four fuzzy systems (FS) are designed and then for other lower levels data is aggregated. Fuzzy expert system at first level has five-linguistic variables and three antecedents; therefore, we have three variables with five linguistic terms. Therefore, we have 5^3 rules. Fuzzy expert system at second level has five-linguistic variables and four antecedents; therefore, we have four variables with five linguistic terms. Therefore, we have 3×5^4 rules. After gathering questionnaires the degree of each component is determined as fuzzy triangular numbers and the FS designed in MATLAB. Then for entering these data as fuzzy triangular numbers to MATLAB, we used Simulink.

This conceptual model was tested at Iran Manufacturing Industry in length of time about three years and was assessed with confirmatory factor analysis [20]. Ceramic tile industry in Iran in the last two decades has grown considerably. In Meybod industrial district, there are 28 companies which produce about 70 percent of total ceramic tile in Iran. In this paper, one of these companies, previously studied in terms of knowledge management, was selected. Questionnaires were distributed in three departments to determine the degree of each component. Then, we distributed another questionnaire among ceramic tile industry's experts to design FS.

5 Conclusion and Suggestions

Table 1 show the final results which are depicted graphically in Figures 5. Degree of agility of company A at pessimistic (L) state is 0.2, at optimistic (U) state is 0.47 and most Possible (M) state is 0.38. This methodology used in this study has got the following advantages over other studies. Unlike the previous researches in which the obtained results are not analyzable due to fact that the value obtained is a crisp,

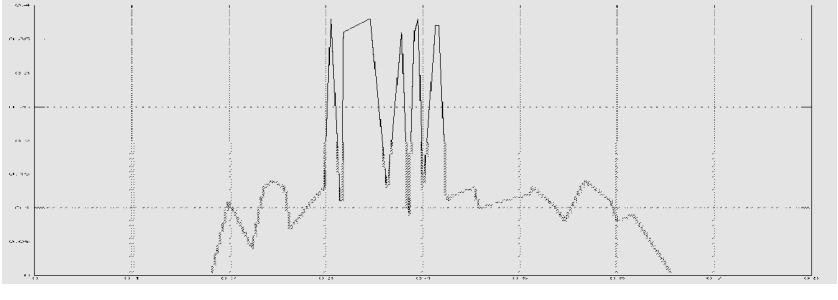


Fig. 5. Degree of agility

the results of present study are more flexible. Because in this study results are presented inform of a range of values which gives the managers a wider range of alternatives for decision making. So far there hasn't been a strategic model to measure agility. One such model has been offered and the present study. The system presented this study clearly shows the strengths and weaknesses of dimensions of agility and its subcomponents. Following the methodology offered in the present study, all the management concepts can be measured by fuzzy system. Future research can divide each component into more subcomponents and devise the respective fuzzy system. Also research in future can be directed towards designing a maturity model for agility.

Table 1. Final results

	Responsiveness			Flexibility			KM		
	(L)	(M)	(U)	(L)	(M)	(U)	(L)	(M)	(U)
People	0.224	0.316	0.433	0.335	0.341	0.394	0.385	0.391	0.424
Process	0.259	0.416	0.616	0.263	0.315	0.563	0.363	0.425	0.593
Strategy	0.182	0.345	0.545	0.236	0.298	0.305	0.286	0.378	0.395
Infrastructure	0.257	0.457	0.657	0.513	0.537	0.567	0.453	0.51	0.513

References

[1] Amit, R.: Schoemaker: Strategic assets and organizational rent. SMJ 14(1), 33–46 (1993)
[2] Berkan, R.Z., Trubatch, S.L.: Fuzzy Systems Design Principles: Building Fuzzy If-Then Rule Bases. Wiley-IEEE Press (1997)
[3] Bourne, M., Mills, J., Wilcox, M., Neely, A., Platts, K.: Designing, implementing and updating performance measurement systems. IJOPM 20(7), 754–771 (2000)
[4] Chan, F.T.S.: Performance measurement in a supply chain. International Journal of Advanced Manufacturing Technology 21, 534–548 (2003)
[5] Chen, S.J., Hwang, C.L.: Fuzzy Multiple Attribute Decision Making: Method and Application. Springer, Berlin (1994)
[6] Cox, E., O'Hagan, M.: The Fuzzy Systems Handbook: A Practitioner's Guide to Building, 2nd edn. Academic Press, London (1999)
[7] McFarlane, D., et al.: A Production Responsiveness Audit. In: Proceedings of POMS, San Francisco, USA (2001)
[8] Dobson, P., Starkey, K.: The Strategic Management Blueprint. Blackwell Pub., Malden (1993)
[9] Dove, R.: Knowledge Management, Response Ability, and The Agile enterprise. Journal of Knowledge Management 3(1), 18–35 (1999)
[10] Dubois, D., Prade, H.: What are fuzzy rules and how to use them? Fuzzy Sets and Systems 84(2), 169–185 (1996)
[11] Dubois, D., Prade, H.: Fundamentals of fuzzy sets. Kluwer Academic Publishers, Dordrecht (2000)
[12] Eisenhardt, K.M., Martin, J.A.: Dynamic capabilities: what are they? SMJ 21, 1105–1121 (2000)
[13] Goldman, S.L., Nagel, R.N., Preiss, K.: Agile competitors and virtual organizations: Strategies for enriching the customer. Van Nostrand, Reinhold, New York (1995)

[14] González, B., Adenso-Díaz, B., González-Torre, P.L.: A fuzzy logic approach for the impact assessment in LCA. Resources, Conservation and Recycling 37, 61–79 (2007)
[15] Henderson, R., Cockburn, I.: Measuring competence? Exploring firm effects in pharmaceutical research. SMJ 15, 63–84 (1994)
[16] http://www.m-w.com
[17] Hwang, C., Lai, Y.: Fuzzy Multi Objective Decision Making: Method and Application. Springer, Berlin (1994)
[18] Itami, H., Roehl, T.: Mobilizing invisible assets. Harvard University Press, Cambridge (1987)
[19] Javidan, M.: Core competence: What dose it mean in practice? LRP 31(1), 60–71 (1998)
[20] Khoshsima, G.: Measuring the agility of manufacturing organizations: Confirmatory and exploratory analysis of a construct, Working Paper
[21] Kogut, B., Zander, U.: Knowledge of the firm, combinative capabilities, and the replication of technology. Organization Science 3, 342–355 (1992)
[22] Kritchanchai, D., McCarthy, B.L.: Developing a set of measures for responsiveness- a survey of the food industry in Thailand. In: Proceeding of POM, San Francisco, April 5-8 (2002)
[23] Kritchanchai, D., McCarthy, B.L.: Responsiveness and strategy in manufacturing. In: Proceedings of IEE workshop Responsiveness in manufacturing, IEE Digest, Savoy place, London 98/2/3
[24] Kritchanchai, D., McCarthy, B.L.: Responsiveness of the order fulfillment process. IJOPM 19(8), 812–833 (1999)
[25] Kuwaiti, M.E., Kay, J.M.: The role of performance measurement in business process reengineering. IJOPM 20(12), 1411–1426 (2000)
[26] Laitinen, E.K.: A dynamic performance measurement system: Evidence from small Finnish technology companies. Scandinavian Journal of Management 18, 65–99 (2002)
[27] Lawrence, P.R., Lorsch, J.W.: Organization and environment: managing differentiation and integration, Irwin, Homewood (1967)
[28] Leonard-Barton, D.: Core capabilities and core rigidities: a paradox in managing new product development. SMJ 13, 111–126 (1992) (Special Issue)
[29] Liang, G.S., Wang, M.J.J.: A Fuzzy Multi-Criteria Decision-Making Approach for Robot Selection. Robotics & Computer-Integrated Manufacturing 10, 267–274 (1993)
[30] Lin, C.T., Chiu, H., Tseng, Y.H.: Agility evaluation using fuzzy logic. IJPE 101, 353–368 (2006)
[31] Long, C., Vickers-Koch, M.: Using core capabilities to create competitive advantage. Organizational Dynamics, 6–22 (Summer 1995)
[32] Matson, J.B., McFarlane, D.C.: Assessing the responsiveness of existing production operations. IJOPM 19(8), 765–784 (1999)
[33] McFarlane, D., Matson, J.: Assessing and improving the responsiveness of manufacturing production systems. In: Proceedings of IEE Workshop on Mass Customization (1999)
[34] Meade, L.M., Sarkis, J.: Analyzing Organizational Project alternatives for agile manufacturing processes: An analytical network approach. IJPR 37(2), 241–261 (1999)
[35] Mintzberg, H., Ahlstrand, B., Lampel, J.: Strategy safari: A guide tour through the wilds of strategic management. Prentice-Hall, Englewood Cliffs (1998)
[36] Neely, A., Gregory, M., Platts, K.: Performance measurement system design: A literature review and research agenda. IJOPM 15(4), 80–116 (1995)
[37] Nelson, R.: Why do firms differ and how does it matter? SMJ 12, 61–74 (1991) (Special Issue)
[38] Next Generation Manufacturing, Agility forum (1997)

[39] Penrose, E.T.: The theory of the growth of the firm. John Willy, New York (1995)
[40] Prahalad, C.K., Hamel, G.: The core competence of the corporation. HBR, 79–91 (1990)
[41] Sharifi, H., Zhang, Z.: A methodology for achieving agility in manufacturing organizations. IJOPM 21(5/6) (2001)
[42] Sharifi, H., Zhang, Z.: A methodology for achieving agility in manufacturing organizations: An introduction. IJPE 62, 7–22 (1999)
[43] Spender, J.C.: The geographics of strategic competence, Mimeo, Rutgers University. In: Prince Bertil Symposium on the Dynamic Firm, Stockholm (1994)
[44] Stalk, G., Evans, P., Shulman, L.E.: Competing on capabilities: The new rules of corporate strategy. HBR, 57–69 (March-April 1992)
[45] Teece, D.J., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management, Mimeo, Hass School of Business, University of California, Berkeley (1992)
[46] Teece, D.J., Pisano, G., Shuen, A.: Dynamic capabilities and strategic management. SMJ 18(7), 509–533 (1997)
[47] Tsourveloudis, N.C., Phillis, Y.A.: Fuzzy assessment of machine flexibility. IEEE Transactions on Engineering Management 45(1), 78–87 (February 1998)
[48] Tsourveloudis, N.C., Phillis, Y.A.: Manufacturing flexibility measurement: A fuzzy logic framework. IEEE Transactions on Robotics and Automation 4(4), 513–524 (1998)
[49] Tsourveloudis, N.C., Phillis, Y.A.: On the Measurement of Enterprise Agility. Journal of Intelligent and Robotic Systems 33, 329–342 (2002)
[50] Ulrich, D., Lake, D.: Organizational capability: Creating competitive advantage. AME 5(1), 77–92 (1991)
[51] Ulrich, D., Wiersema, M.F.: Gaining strategic organizational capability in a turbulent business environment. The Academy of Management Executive 3(2), 115–122 (1989)
[52] Van Hoek, R.I.: Measuring the unmeasurable-measuring and improving performance in the supply chain. Supply Chain Management 3(4), 187–192 (1998)
[53] Rothwell, W.J., Lindholm, J.E.: Competency identification, modeling and assessment in the USA. IJOTD 3(2), 90–105 (1999)
[54] Waggoner, D.B., Neely, A.D., Kennerley, M.P.: The forces that shape organizational performance measurement systems: An interdisciplinary review. IJPE 60, 53–60 (1999)
[55] Yusuf, Y.Y., et al.: Agile Manufacturing: the Drivers, Concepts and Attributes. International Journal of Production Economics 62, 33–43 (1999)
[56] Zadeh, L.A.: Fuzzy sets, Information and Control 8(3), 338–353 (1965)

Comparison on Gradient-Based Neural Dynamics and Zhang Neural Dynamics for Online Solution of Nonlinear Equations

Yunong Zhang, Chenfu Yi, and Weimu Ma

School of Information Science and Technology, Sun Yat-Sen University
Guangzhou 510275, China
zhynong@mail.sysu.edu.cn, ynzhang@ieee.org
http://www.ee.sysu.edu.cn/teacher/detail.asp?sn=129

Abstract. For online solution of nonlinear equation $f(x) = 0$, this paper generalizes a special kind of recurrent neural dynamics by using a recent design method proposed by Zhang *et al*. Different from gradient-based dynamics (GD), the resultant Zhang dynamics (ZD) is designed based on the elimination of an indefinite error-monitoring function (instead of the elimination of a square-based positive error-function usually associated with GD). For comparative purposes, the gradient-based dynamics is also developed and exploited for solving online such a nonlinear equation $f(x) = 0$. Computer-simulation results via power-sigmoid activation functions substantiate further the theoretical analysis and efficacy of Zhang neural dynamics on nonlinear equations solving.

Keywords: Recurrent neural networks, neural dynamics, nonlinear equations, activation functions, exponential convergence.

1 Introduction

The solution of nonlinear equations in the form of $f(x) = 0$ is widely encountered in science and engineering fields. Many computational methods and models have thus been developed. For example, numerical algorithms are employed popularly for solving such nonlinear equations [1,2,3]. However, it may not be efficient enough for most numerical algorithms because of their serial-processing nature performed on digital computers [4]. Recently, due to the in-depth research in neural networks, the dynamic-system approach using recurrent neural models has become one of the important parallel-processing methods for solving online optimization and algebraic problems [5,6,7,8,9,10,11,12,13,14,15,16,17]. Generally speaking, most reported computational-schemes are related to the gradient-descent method or other methods intrinsically designed for constant problems (or to say, time-invariant problems, static problems, or stationary problems) solving.

Different from gradient-based neural-dynamic approach [5,6,7,8,18], a special kind of recurrent neural dynamics has recently been proposed by Zhang *et al*

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 269–279, 2008.
© Springer-Verlag Berlin Heidelberg 2008

[7,9,10] for time-varying Sylvester equation solving and time-varying matrix inversion. By following and generalizing Zhang *et al*'s design method, a neural dynamics is elegantly introduced in this paper to solve nonlinear equations. The resultant Zhang dynamics (ZD, termed and abbreviated as such for presentation convenience) is developed by defining an indefinite error-monitoring function and then making it exponentially decrease to zero. For comparative purposes, the conventional gradient-based neural dynamics is investigated as well in this paper. Theoretical and simulative results both show the effectiveness, accuracy and efficiency of the resultant Zhang neural dynamics on equations solving.

2 Problem Formulation and Neural-Dynamic Solvers

Our objective in this paper is to find solution $x \in \mathbb{R}$ so as to make the following solvable nonlinear equation hold true:

$$f(x) = 0. \tag{1}$$

For discussion and comparison purposes, x^* denotes a theoretical solution to (1). The design procedures of gradient-based neural dynamics and Zhang neural dynamics are investigated in this section as follows. In addition, some basic types of activation functions (such as linear, sigmoid, power, and power-sigmoid activation-functions) are analyzed for the convergence of the neural dynamics.

2.1 Gradient-Based Neural-Dynamics

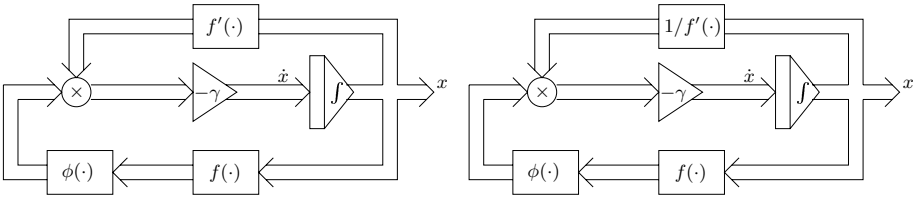
The conventional neural-dynamic computational schemes generally belong to or relate to the gradient-descent method in optimization [5,6,7,8,18]. By such a design method, we could develop the gradient-based neural dynamics for solving the nonlinear equation $f(x) = 0$ as follows. Firstly, to solve nonlinear equation (1), the gradient-descent method requires us to define a square-based positive error-function (or termed, energy function) such as $\mathcal{E}(x) = f^2(x)$. Then, a typical continuous-time adaptation rule based on the negative-gradient information leads to the following differential equation (which we term as gradient dynamics):

$$\frac{dx(t)}{dt} = -\frac{\gamma}{2}\frac{\partial \mathcal{E}(x)}{\partial x} = -\gamma f(x)\frac{\partial f(x)}{\partial x} := -\gamma f(x) f'(x), \tag{2}$$

where γ is a positive design-parameter used to adjust the convergence rate of the neural-dynamics, and $x(t)$, starting from randomly-generated initial condition $x(0) = x_0 \in \mathbb{R}$, is the neural-activation state corresponding to theoretical solution x^*. As an extension to the above design approach and as inspired by [10], we could have the following generalized nonlinear form of gradient-based dynamics (GD) by using nonlinear activation function $\phi(\cdot)$:

$$\dot{x}(t) = -\gamma \phi\big(f(x)\big) f'(x). \tag{3}$$

In addition, it is worth mentioning that, similar to usual neural-dynamic approaches, design parameter γ in (2) and hereafter, being the reciprocal of a



(a) Gradient dynamics (3) (b) Zhang dynamics (5)

Fig. 1. Block diagrams of neural-dynamic solvers for online solution of $f(x) = 0$

capacitance parameter, could be set as large as hardware permits (e.g., in analog circuits or VLSI [7]) or selected appropriately (e.g., between 10^3 and 10^8) for experimental and/or simulative purposes.

2.2 Zhang Neural-Dynamics

By Zhang *et al*'s design method [7,9,10], we could define directly the following indefinite error function (where "indefinite" means that the function could be positive, zero, negative, and bounded or even unbounded) so as to generalize a Zhang-dynamic model for solving online the nonlinear equation $f(x) = 0$:

$$e(t) := f\big(x(t)\big).$$

Then, the time derivative $\dot{e}(t)$ of error-function $e(t)$ could be chosen and forced mathematically such that the error function $e(t)$ exponentially converges to zero. Specifically, $\dot{e}(t)$ can be chosen in the following general form:

$$\frac{de(t)}{dt} = -\gamma\phi\big(e(t)\big),$$

or equivalently,

$$\frac{df(x)}{dt} = -\gamma\phi\big(f(x)\big), \tag{4}$$

where design-parameter $\gamma > 0$ and activation-function $\phi(\cdot) : \mathbb{R} \to \mathbb{R}$ are defined as before. Expanding the above ZD design formula, we could thus obtain

$$\frac{\partial f(x)}{\partial x}\frac{dx}{dt} = -\gamma\phi\big(f(x)\big),$$

leading to the following differential equation (which we term Zhang dynamics):

$$f'(x)\dot{x}(t) = -\gamma\phi\big(f(x)\big) \quad \text{and/or} \quad \dot{x}(t) = -\gamma\phi\big(f(x)\big)/f'(x), \tag{5}$$

where $x(t)$, starting from randomly-generated initial condition $x(0) = x_0 \in \mathbb{R}$, is the neural-activation state corresponding to theoretical solution x^* of (1). In addition, it is worth pointing out that a general form of Newton-Raphson

iterative method [19] for solving nonlinear equation $f(x)$=0 might be derived by discretizing this Zhang neural-dynamics instead of gradient neural-dynamics.

Moreover, the block-diagram representation of neural-dynamics (3) and (5) could be shown in Fig. 1. In view of equations (3), (5) and Fig. 1, different choices for γ and $\phi(\cdot)$ may lead to different performance of the neural-dynamics. In general, any monotonically-increasing odd activation function $\phi(\cdot)$ could be used for the construction of the neural dynamics. For example, the following four basic types of activation function $\phi(\cdot)$ could be adopted in this paper:

1) linear activation function $\phi(u) = u$,
2) bipolar sigmoid activation function (with $\xi > 2$)

$$\phi(u) = \frac{1 - \exp(-\xi u)}{1 + \exp(-\xi u)},$$

3) power activation function $\phi(u) = u^p$ with odd integer $p \geqslant 3$ (note that linear activation function $\phi(u) = u$ can be viewed as a special case of power activation function with power-index $p = 1$), and
4) power-sigmoid activation function

$$\phi(u) = \begin{cases} u^p, & \text{if } |u| \geqslant 1 \\ \frac{1+\exp(-\xi)}{1-\exp(-\xi)} \cdot \frac{1-\exp(-\xi u)}{1+\exp(-\xi u)}, & \text{otherwise} \end{cases} \tag{6}$$

with suitable design parameters $\xi \geqslant 1$ and $p \geqslant 3$.

It is worth saying that other types of activation functions can be generalized and extended by understanding the above four basic types of activation functions.

3 Theoretical Analysis

While Section 2 presents the general frameworks about gradient-based dynamics and Zhang dynamics for solving nonlinear equation $f(x) = 0$, detailed design consideration and theoretical results are given in this section. To analyze the convergence of Zhang dynamics (5), we firstly introduce the following definitions.

Definition 1. A neural-network system is said to be globally convergent, if starting from any initial state taken in the whole associated Euclidean space, every state trajectory of the neural network converges to an equilibrium point that may depend on the initial state of the system trajectory [20].

Definition 2. A neural-network system is said to be globally exponentially convergent, if every trajectory starting from any initial point $x(t_0)$ satisfies

$$\|x(t) - x^*\| \leqslant \eta \|x(t_0) - x^*\| \exp\big(-\lambda(t - t_0)\big), \ \forall t \geqslant t_0 \geqslant 0,$$

where constants $\eta > 0$ and $\lambda > 0$ exist, x^* denotes here an equilibrium point that may depend on initial state $x(t_0)$, and symbol $\|\cdot\|$ denotes the Euclidean

norm for a vector (which, in our present situation, denotes the absolute value of a scalar argument) [20].

For Zhang dynamics (5), we could have the following propositions.

Proposition 1. Consider a solvable nonlinear equation $f(x) = 0$, where $f(\cdot)$ is a continuously differentiable function. If a monotonically-increasing odd activation function $\phi(\cdot)$ is employed, then neural state $x(t)$ of Zhang dynamics (5) starting from randomly-generated initial state $x(0) = x_0 \in \mathbb{R}$ could converge to theoretical solution x^* of nonlinear equation $f(x) = 0$ depicted in (1).

Proof. We can define a Lyapunov function candidate $V(x) = f^2(x)/2 \geqslant 0$, and its time derivative along the system trajectory of Zhang dynamics (5) becomes

$$\frac{dV(x)}{dt} = f(x)f'(x)\frac{dx}{dt} = -\gamma f(x)\phi\big(f(x)\big). \tag{7}$$

Because a monotonically-increasing odd function is used as an activation function, we could have $\phi\big(-f(x)\big) = -\phi\big(f(x)\big)$, and

$$\phi\big(f(x)\big)\begin{cases} > 0, & \text{if } f(x) > 0, \\ = 0, & \text{if } f(x) = 0, \\ < 0, & \text{if } f(x) < 0. \end{cases} \tag{8}$$

Hence we have

$$f(x)\phi\big(f(x)\big)\begin{cases} > 0, & \text{if } f(x) \neq 0, \\ = 0, & \text{if } f(x) = 0, \end{cases} \tag{9}$$

which guarantees the final negative-definiteness of $\dot{V}(x)$; i.e., $\dot{V}(x) < 0$ for $f(x) \neq 0$ (equivalently, $x \neq x^*$) and $\dot{V}(x) = 0$ for $f(x) = 0$ (equivalently, $x = x^*$). By Lyapunov theory [20], residual error $e(t) = f(x(t))$ could converge to zero. That is, neural state $x(t)$ of ZD (5) could converge to a theoretical solution x^* with $f(x^*) = 0$ starting from some randomly-generated initial states [note that $f'(x)$ appears in the derivation of (7)]. The proof is now complete.

Proposition 2. Let x^* denote a theoretical solution to problem $f(x) = 0$, where $f(\cdot)$ is a continuously-differentiable function (specifically, with at least first-order derivatives at some interval containing x^*). In addition to Proposition 1, neural state $x(t)$ of Zhang dynamics (5) could converge to x^* if initial state x_0 is close enough to x^*. Moreover, Zhang dynamics (5) possesses the following properties.

1) If the linear activation function is used, then exponential convergence with rate γ [in terms of residual error $e(t) = f(x(t))$] can be achieved for (5).
2) If the bipolar-sigmoid activation-function is used, superior convergence can be achieved for error range $e(t) = f(x(t)) \in [-\delta, \delta]$, $\exists \delta > 0$, as compared to the situation of using the linear activation function described in Property 1.
3) If the power activation function is used, then superior convergence can be achieved for error ranges $(-\infty, -1)$ and $(1, +\infty)$, as compared to the situation of using the linear activation function described in Property 1.

4) If the power-sigmoid activation function is used, superior convergence can be achieved for the whole error range $e(t) = f(x(t)) \in (-\infty, +\infty)$, as compared to the situation of using the linear activation function in Property 1.

Proof. We now come to prove the additional convergence properties of Zhang dynamics (5) by using the mentioned several types of activation functions $\phi(\cdot)$.

1) For the situation of using the linear activation function, it follows from equation (4) that $df(x)/dt = -\gamma f(x)$, which yields $f(x(t)) = \exp(-\gamma t)f(x_0)$. This proves the exponential convergence rate γ of Zhang dynamics (5) in the sense of its residual error $f(x(t)) \to 0$. Furthermore, we could also show the state convergence of Zhang dynamics to theoretical solution x^* via the following procedure. According to Taylor's theorem [19], we could have

$$\begin{aligned} f(x) =& f(x - x^* + x^*) \\ =& f(x^*) + (x - x^*)f'(x)|_{x=x^*} + \frac{(x-x^*)^2}{2!}f''(x)|_{x=x^*} + \cdots \\ &+ \frac{(x-x^*)^n}{n!}f^{(n)}(x)|_{x=x^*} + \frac{(x-x^*)^{n+1}}{(n+1)!}f^{(n+1)}(x)|_{x=\alpha}, \end{aligned}$$

for some α existing between $x(t)$ and x^*. Thus, in view of $f(x(t))=\exp(-\gamma t)f(x_0)$ and $f(x^*)=0$, by omitting higher-order terms, we could have $(x-x^*)f'(x)|_{x=x^*} \approx \exp(-\gamma t)f(x_0)$, which, if $f'(x)|_{x=x^*} \neq 0$, yields

$$\|x - x^*\| \approx \exp(-\gamma t)f(x_0)/f'(x)|_{x=x^*}$$

and shows the exponential convergence of Zhang dynamics (5) in terms of neural state $x(t) \to x^*$ as well. In addition, note that, even if $f^{(k)}(x)|_{x=x^*} = 0$, $\forall k = 1, 2, \cdots, (n-1)$, and $f^{(n)}(x)|_{x=x^*} \neq 0$, we still have

$$\|x - x^*\| \approx \exp(-\gamma t/n)\big(f(x_0)n!/f^n(x)|_{x=x^*}\big)^{1/n},$$

which shows again the exponential convergence of Zhang dynamics (5) in terms of neural state $x(t) \to x^*$, provided that initial state x_0 is close enough to x^*.

2) For the situation of using bipolar-sigmoid function $\phi(u) = \big(1-\exp(-\xi u)\big)/\big(1+\exp(-\xi u)\big)$, we know that there exists an error range $e(t) = f(x(t)) \in [-\delta, \delta]$ with $\delta > 0$ such that $\|\phi(f(x))\| > \|f(x)\|$ [7,10]. So, by reviewing the proof of Proposition 1 [especially, equation (7)], we know further that superior convergence can be achieved by ZD (5) with bipolar-sigmoid activation function for such an error range, as compared to Property 1 of using the linear function.

3) For the p-th power activation function $\phi(u) = u^p$, it follows from (4) that $df(x(t))/dt = -\gamma f^p(t)$, and its general form of trajectory is written down as

$$f(x(t)) = f(x_0)\big((p-1)f^{p-1}(x_0)\gamma t + 1\big)^{-\frac{1}{p-1}}.$$

Specifically, for $p = 3$, residual error $f(x(t)) = f(x_0)/\sqrt{2f^2(x_0)\gamma t + 1}$. Evidently, as $t \to \infty$, $f(x) \to 0$. Look back to the proof of Proposition 1 [specifically,

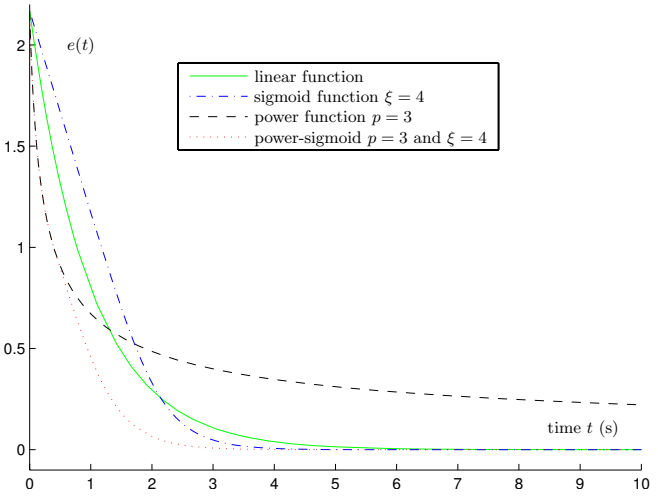


Fig. 2. Convergence behavior of error $e(t)$ with different activation functions

equation (7)]: as for Lyapunov function candidate $V(x) = f^2(x)/2$ and its time derivative $\dot{V}(x) = -\gamma f(x)\phi(f(x))$, we could have $f(x)\phi(f(x)) = f^{p+1}(x) \gg f^2(x)$ for error ranges $f(x) \ll -1$ and/or $f(x) \gg 1$. This implies that when using the power activation function, a much faster convergence can be achieved by ZD (5) for such error ranges in comparison with Property 1 [7,10].

4) It follows from the above analysis (especially, Properties 2 and 3) that, in order to achieve superior convergence, a high-performance neural dynamics could be developed by switching the power activation function to the bipolar-sigmoid activation function at switching points $f(x) = \pm 1$. Thus, if the power-sigmoid activation function is used with suitable design parameters $\xi \geqslant 1$ and $p \geqslant 3$, superior convergence could be achieved for ZD (5) theoretically over the whole error range $(-\infty, +\infty)$, as compared to the linear-activation-function situation.

For graphical interpretation, the convergence behavior of residual error $e(t)$ is illustrated in Fig. 2 by using different activation functions, where $\gamma = 1$. Note that, to draw all curves in the same one plot, small values of design parameters of nonlinear activation functions (such as $\xi = 4$ and $p = 3$) have to be used.

Remark 1. Nonlinearity always exists, which is one of the main motivations for us to investigate different activation functions in this kind of work. Even if the linear activation function is used, nonlinear phenomenon may still appear in its hardware implementation; e.g., in the form of saturation and/or inconsistency of linear slope, and in digital realization due to truncation and round-off errors [7]. The investigation of different activation functions (such as the sigmoid and power functions) may give us more insights into the imprecision problem and side-effects possibly appearing in neural hardware implementation.

Remark 2. It is worth comparing here the two design methods of neural dynamics; namely, Zhang dynamics (5) and gradient-based dynamics (3). They are both exploited for online solution of nonlinear equation (1). However, gradient-based dynamics (3) is designed based on the elimination of square-based error-function $\varepsilon(x) = f^2(x)$ as well as gradient-descent method. In contrast, Zhang dynamics (5) is designed based on the elimination of an indefinite error-function $e(t) = f(x)$ itself, which might be positive, negative, bounded or unbounded.

4 Computer-Simulation Studies

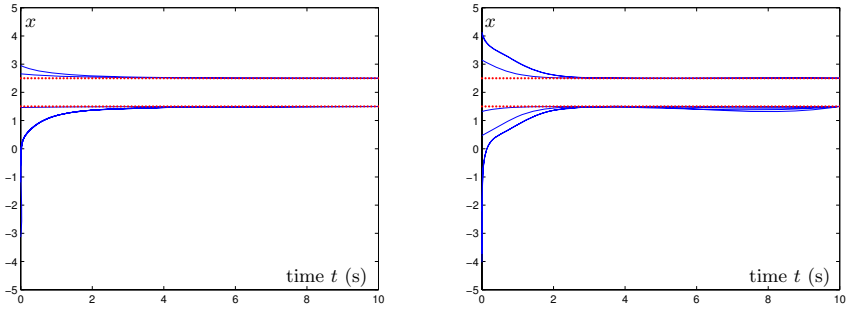
While Sections 2 and 3 present gradient-based neural dynamics (3), Zhang neural dynamics (5) and related analysis results, in this section we show computer-simulation results so as to demonstrate the characteristics of the neural-dynamic convergence. For comparative purposes, both dynamics are exploited for solving online the nonlinear equation $f(x) = 0$ with two illustrative examples below.

Example 1. Consider the following nonlinear equation in the quadratic form:

$$f(x) = x^2/2 - 2x + 1.875 = 0. \tag{10}$$

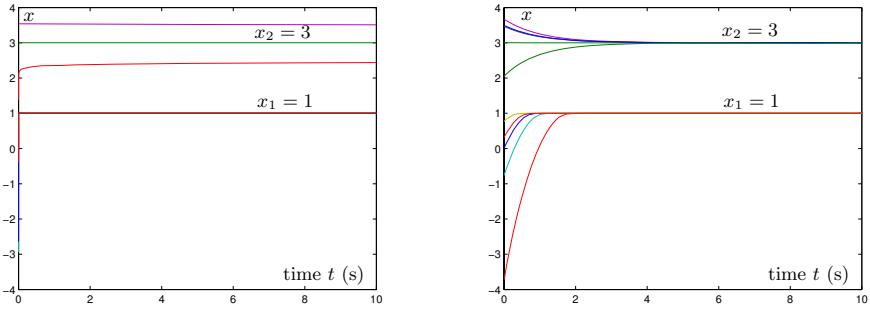
For comparative purposes, the theoretical solutions to the above nonlinear equation could be written down as $x_1^* = 1.5$ and $x_2^* = 2.5$. These are used here to verify the theoretical results discussed in the previous sections.

As seen from Fig. 3, starting from randomly-generated initial states selected within $[-5, 5]$, the activation state $x(t)$ of the investigated neural-dynamics (3) and (5) could both converge to a theoretical solution, either x_1^* or x_2^* as denoted by dotted lines in red.



(a) Gradient dynamics (3) (b) Zhang dynamics (5)

Fig. 3. Online solution of nonlinear equation (10) by randomly-initialized GD (3) and ZD (5) with $\gamma = 1$, where dotted lines in red denote the theoretical solutions



(a) Gradient dynamics (3) (b) Zhang dynamics (5)

Fig. 4. Online solution of (11) by randomly-initialized GD (3) and ZD (5) with $\gamma = 10$

Example 2. Consider the following nonlinear equation with multiple roots:

$$f(x) = (x-3)^{10}(x-1) = 0. \tag{11}$$

Evidently, the theoretical solutions to the above nonlinear equation could be written simply as $x_1^* = 1$ (a simple root) and $x_2^* = 3$ (a root of order 10). As seen from Fig. 4(a), by applying gradient-based neural dynamics (3), starting from randomly-generated initial states selected within $[-4, 4]$, some wrong solutions (other than $x_1^* = 1$ and $x_2^* = 3$) have also been yielded. In contrast, neural state $x(t)$ of the presented Zhang dynamics (5) could converge to a theoretical solution in this example, either $x_1^* = 1$ or $x_2^* = 3$; i.e., no wrong solution synthesized by Zhang dynamics! This could be seen clearly from Fig. 4(b).

In summary, gradient dynamics (3) could effectively solve some nonlinear equations (e.g., Example 1). However, it might yield some inapplicable solutions (e.g., Example 2) due to the multiplication of the derivative term in its dynamics. In contrast, as proved theoretically and substantiated simulatively in this paper, Zhang dynamics (5) could more accurately and effectively solve the nonlinear equation $f(x) = 0$ depicted in (1) if there exists a solution to it.

5 Conclusions

A special kind of neural dynamics has been generalized, developed, analyzed and compared in this paper by following Zhang *et al*'s method for online solution of nonlinear equations. Different from conventional gradient-based neural dynamics, such a Zhang neural dynamics has been elegantly introduced by defining an indefinite error-monitoring function (instead of usually-exploited positive energy function). Thus, its computational error could be made exponentially decrease to zero. For comparative purposes, the gradient-based neural dynamics has also been applied to online solution of such nonlinear equations. Theoretical and simulative results have both demonstrated the accuracy and efficacy of Zhang

dynamic approach. Further research efforts may be directed towards the design, development and analysis of discrete-time neural models.

References

1. Yang, L., Zhu, Z., Wang, Y.: Exact Solutions of Nonlinear Equations. Physics Letters A 260, 55–59 (1999)
2. Babolian, E., Biazar, J.: Solution of Nonlinear Equations by Modified Adomian Decomposition Method. Applied Mathematics and Computation 132, 167–172 (2002)
3. Babolian, E., Biazar, J., Vahidi, A.R.: Solution of a System of Nonlinear Equations by Adomian Decomposition Method. Applied Mathematics and Computation 150, 847–854 (2004)
4. Zhang, Y., Leithead, W.E., Leith, D.J.: Time-Series Gaussian Process Regression Based on Toeplitz Computation of $O(N^2)$ Operations and $O(N)$-level Storage. In: Proceedings of IEEE Conference on Decision and Control, pp. 3711–3716 (2005)
5. Jang, J., Lee, S., Shin, S.: An Optimization Network for Matrix Inversion. In: Anderson, D.Z. (ed.) Neural Information Processing Systems, pp. 397–401. American Institute of Physics, New York (1998)
6. Wang, J.: A Recurrent Neural Network for Real-Time Matrix Inversion. Applied Mathematics and Computation 55, 89–100 (1993)
7. Zhang, Y.: Revisit the Analog Computer and Gradient-Based Neural System for Matrix Inversion. In: Proceedings of IEEE International Symposium on Intelligent Control, pp. 1411–1416 (2005)
8. Zhang, Y.: Towards Piecewise-Linear Primal Neural Networks for Optimization and Redundant Robotics. In: Proceedings of IEEE International Conference on Networking, Sensing and Control, pp. 374–379 (2006)
9. Zhang, Y., Jiang, D., Wang, J.: A Recurrent Neural Network for Solving Sylvester Equation with Time-Varying Coefficients. IEEE Transactions on Neural Networks 13, 1053–1063 (2002)
10. Zhang, Y., Ge, S.S.: Design and Analysis of a General Recurrent Neural Network Model for Time-Varying Matrix Inversion. IEEE Transactions on Neural Networks 16, 1477–1490 (2005)
11. Zhang, Y., Wang, J.: Global Exponential Stability of Recurrent Neural Networks for Synthesizing Linear Feedback Control Systems via Pole Assignment. IEEE Transactions on Neural Networks 13, 633–644 (2002)
12. Zhang, Y., Wang, J.: A Dual Neural Network for Convex Quadratic Programming Subject to Linear Equality and Inequality Constraints. Physics Letters A 298, 271–278 (2002)
13. Zhang, Y.: A Set of Nonlinear Equations and Inequalities Arising in Robotics and its Online Solution via a Primal Neural Network. Neurocomputing 70, 513–524 (2006)
14. Steriti, R.J., Fiddy, M.A.: Regularized Image Reconstruction Using SVD and a Neural Network Method for Matrix Inversion. IEEE Transactions on Signal Processing 41, 3074–3077 (1993)
15. Tank, D., Hopfield, J.: Simple Neural Optimization Networks: an A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit. IEEE Transactions on Circuits and Systems 33, 533–541 (1986)
16. Cichocki, A., Unbehauen, R.: Neural Network for Solving Systems of Linear Equations and Related Problems. IEEE Transactions on Circuits and Systems 39, 124–138 (1992)

17. Cichocki, A., Unbehauen, R.: Neural Networks for Optimization and Signal Processing. Wiley, Chichester (1993)
18. Manherz, R.K., Jordan, B.W., Hakimi, S.L.: Analog Methods for Computation of the Generalized Inverse. IEEE Transactions on Automatic Control 13, 582–585 (1968)
19. Mathews, J.H., Fink, K.D.: Numerical Methods Using MATLAB, 4th edn. Pretice Hall, New Jersey (2004)
20. Zhang, Y.: Dual Neural Networks: Design, Analysis, and Application to Redundant Robotics. In: Kang, G.B. (ed.) Progress in Neurocomputing Research, pp. 41–81. Nova Science Publishers, New York (2007)

Competitive Hopfield Neural Network with Periodic Stochastic Dynamics for Partitional Clustering

Zhanghui Kuang, Wei Bi, and Jiahai Wang*

Department of Computer Science, Sun Yat-sen University, No. 135,
Xingang West Road, Guangzhou 510275, PR China
kuangzhh@mail2.sysu.edu.cn,
wangjiah@mail.sysu.edu.cn

Abstract. A novel competitive Hopfield network with periodic stochastic dynamics is proposed for the NP-hard partitional clustering problem in this paper. Clustering technique has been applied to a wide range of problems, such as pattern recognition and machine learning. The aim of partitional clustering is to obtain a specified number of data sets from the original data according to certain criteria. The proposed algorithm introduces periodic stochastic dynamics, which helps the neural network escape from local minima and search a possible better solution based on the solution which is obtained in the latest period. The performance is evaluated through several benchmark data sets. The simulation results show that the proposed algorithm outperforms previous approaches, such as k-means, genetic algorithm, particle swarm optimization, differential evolution, combinatorial particle swarm optimization and tabu search.

Keywords: Partitional Clustering, Competitive Hopfield Neural Network, Periodic Stochastic Dynamics.

1 Introduction

Data clustering is utilized to find natural groupings in data. It acts a central role in variety of fields. Clustering techniques have been used to a wide range of pattern recognition and machine learning. A many of clustering algorithms have been proposed in last several decades [1]. Most of the algorithms can be classified into two types: hierarchical clustering and partitional clustering [1]. Partitional clustering algorithms group a data set into a specified number of data subsets according to certain criteria, e.g. a square error function. This paper mainly focuses on partitional clustering.

A lot of algorithms have been introduced in the literatures to solve the partitional clustering problem [1], [2]. Forgy [2] proposed an important one, called k-means algorithm. However, it is sensitive to the initial selection of centers and is likely to converge to partitions which are significantly inferior to the global optimum. Bhuyan, et al. [3], [4] applied genetic algorithm (GA) to find globally optimal solutions to the clustering problems with high computational complex. Recently, Paterlini et al. [5]

* Corresponding author.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 280–289, 2008.
© Springer-Verlag Berlin Heidelberg 2008

proposed two algorithms, particle swarm optimization (PSO) and differential evolution (DE), compared with k-means, random search (RS), and GA, to solve the partitional clustering problem. Further, Jarboui, et al. [6] proposed a combinatorial particle swarm optimization (CPSO) and a GA for the partitional clustering. Liu et al. [7] designed a new tabu search called TS-Clustering to deal with the clustering problem.

Since the partitional clustering problem can be considered as optimization problems, Hopfield neural networks [8] can be used to solve it. Recently, Galán-Marín, et al. [9] proposed an optimal competitive Hopfield model (OCHOM), which guarantees and maximizes the descent of any Lyapunov energy function as the groups of neurons are updated. Their simulation results for the bipartite subgraph problem and N-queen problem illustrated that the OCHOM was much superior to other Hopfield neural networks in terms of the solution quality and the computation time.

However, the OCHOM is easily trapped in local minima. This paper proposed a novel algorithm which employs a periodic stochastic dynamics, resulting in periodic temporary energy increasing, to help the OCHOM escape from local minima. Further, the proposed algorithm can seek for a possible better solution based on the solution which is obtained in the latest period. The algorithm is called competitive Hopfield neural network with periodic stochastic dynamics (CHNN-PSD). The performance of the CHNN-PSD is evaluated through several benchmark data sets. The experiments results show that the CHNN-PSD outperforms previous approaches, such as k-means, GA, PSO, DE, CPSO and TS-Clustering for the partitional clustering problem.

The remaining sections of this paper are organized as follows. Section 2 defines the clustering problem. The CHNN-PSD is proposed in section 3. In section 4, we apply the CHNN-PSD to the partitional clustering problem. In Section 5, benchmark data sets are used to evaluate the proposed algorithm. Finally, some conclusions are drawn in last section.

2 Clustering Problem

The clustering problem can be formulated as follows. Let n denote the number of objects or patterns in a D-dimensions space, which will be clustered into K groups or clusters according to certain criteria. If the clusters are represented by $G = \{g_1, g_2, \cdots, g_K\}$, and $n_k = |g_k|$ is the cardinal of the cluster g_k, then these variables must meet constraints: $g_k \neq \phi$, $g_i \bigcap g_j = \phi \quad i \neq j. \quad k, i, j = 1, 2, \cdots K$ and $\sum_{k=1}^{K} n_k = n$. Let $f(G)$ be the objective function of the clustering problem. The objective of the problem is to optimize $f(G)$.

The criteria, which is used widely in partitional clustering, is squared error (SE) criterion. The SE for the partition is the sum of squared Euclidean distance between each object and its cluster center. The SE for the partition $G = \{g_1, g_2, \cdots, g_k\}$ can be calculated as:

$$\mathrm{SE}(G) = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_{ik} - c_k)^T (x_{ik} - c_k) \tag{1}$$

Where x_{ik} is the i th object belonging to the cluster g_k and c_k is the mean vector of cluster g_k. c_k can be described as follows: $c_k = (\sum_{i=1}^{n_k} x_{ik}) / n_k$ .So the objective of the partitional clustering problem is: $\text{minimize SE}(G)$.

Another criterion which is used commonly is a variance ratio criterion (VRC) [6]. At first we define SEinter(G), as a sum of squared distances between the cluster centers c_k and the mean vector of all objects (the center of all objects) c_G, as follows:

$$\text{SEinter}(G) = \sum_{k=1}^{K} n_k (c_k - c_G)^T (c_k - c_G)$$

Where the mean vector of all objects (the center of all objects) c_G is

$$c_G = (\sum_{i=1}^{n} x_i) / n$$

Then, the VRC can be defined as:

$$\text{VRC}(G) = \frac{\text{SEinter}(G) / (K-1)}{\text{SE}(G) / (n-K)} \tag{2}$$

Where $(n-K)$ is the degree of freedom of the $\text{SEinter}(G)$, and $K-1$ is the degree of freedom of the $\text{SE}(G)$.

3 CHNN-PSD

In this section, a novel competitive Hopfield network with periodic stochastic dynamics will be proposed. Below we briefly review the OCHOM model firstly, and then the CHNN-PSD is introduced.

The OCHOM [9] is consisted of K disjoint groups of neurons, where each group of neurons is composed of m neurons. Let $v_{xi}(t) \in \{0,1\}$, $u_{xi}(t)$,and θ_{xi} be the output, input and basis of the i th neuron in the x th group, respectively, where t is discrete time. The interconnection strength between the neuron xi and the neuron yj is denoted by $\omega_{xi,yj}(x, y = 1,2,\cdots K,\ i, j = 1,2,\cdots m)$.

The OCHOM is characterized by the Hopfield energy function:

$$E(t) = -\frac{1}{2}\sum_{x=1}^{K}\sum_{i=1}^{m}\sum_{y=1}^{K}\sum_{j=1}^{m}\omega_{xi,yj}v_{xi}v_{yj} + \sum_{x=1}^{K}\sum_{i=1}^{m}\theta_{xi}v_{xi} \tag{3}$$

The inputs of the neurons are calculated by the discrete Hopfield updating rule:

$$u_{xi} = -\frac{\partial E}{\partial v_{xi}} = \sum_{y=1}^{K}\sum_{j=1}^{m}\omega_{xi,yj}v_{yj} - \theta_{xi} \tag{4}$$

In the OCHOM, the iteration process employs a special rule, called winner-takes-all. One and only one neuron in each group is fired at every time t. Which neuron in the group x is fired with the output 1 at time $t+1$ follows the rule blow.

$$v_{xi}(t+1)=\begin{cases}1 & u_{xi}(t)-K_{x0,xi}=\max\limits_{j=1,\cdots m}(u_{xj}(t)-K_{x0,xj})\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

Where $x0$ denotes the fired neuron for group x at time t and the $K_{x0,xj}$ can be computed as follows:

$$K_{x0,xj}=-\frac{1}{2}(\omega_{x0,x0}+\omega_{xj,xj}-2\omega_{x0,xj}) \tag{6}$$

The update rule of the OCHOM can ensure the energy descent for every time step. The proof can be found in [9].

However, the OCHOM has no mechanism to escape from local minima. In order to solve the local minima possibly, we propose a new algorithm, called CHNN-PSD. The CHNN-PSD incorporates periodic stochastic dynamics into the OCHOM, so that temporary energy ascending is permitted, which helps the OCHOM escape from the local minima. Furthermore, the mechanism can enable the network to search a possible better result based on the best solution which is obtained in the latest period. The input/output function of the i th neuron in the x th group is modified and is such:

$$v_{xi}(t+1)=\begin{cases}1 & u'_{xi}(t)=\max\limits_{j=1,\cdots m}(u'_{xj}(t))\\ 0 & \text{otherwise}\end{cases} \tag{7}$$

$$u'_{xi}=\alpha(t)\cdot(u_{xi}(t)-K_{x0,xi}) \tag{8}$$

Where u'_{xi} is the transient variable. The CHNN-PSD dynamics includes the periodic stochastic dynamics, which is different from the origin OCHOM optimal dynamics. Thus, steepest descent and ascent are achieved at $\alpha(t)=1$ and $\alpha(t)=-1$ respectively. At $\alpha(t)=1$, the dynamics of the CHNN-PSD is the same as those of the OCHOM. At $\alpha(t)=-1$, the neuron with the minimum value of $\{u_{xi}(t)-K_{x0,xi}\}$ for each group x is selected. The cooling schedule for $\alpha(t)$ is given by:

$$\alpha(t)=\text{random}(h(t),1) \tag{9}$$

$$h(t)=1-2e^{-t \bmod(T)/\lambda} \tag{10}$$

Where T is the period of the dynamics function and λ is a parameter to accommodate the speed of the hill-climbing. Fig. 1 shows how the value of h changes with t for $\lambda=50$ and $T=300$.

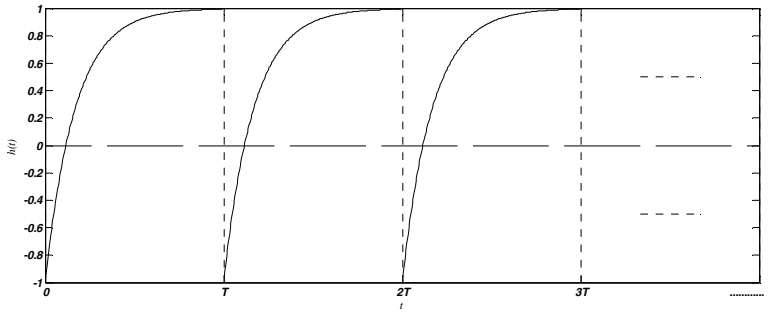


Fig. 1. Graph of $h(t)$ with $\lambda = 50$ and $T = 300$

Initially, $h(0) = -1$, therefore $\alpha(0)$ is a random value that is selected from a range $[-1,1]$. As t increases in a period T, the range of the random is becoming narrower and narrower. When t is enough large, $\alpha(t) \to 1$ is guaranteed. Note that $h(t)$ is a periodic function, the values of which in $[(k-1)T, kT]$ ($k = 1,2,\cdots,n$) are the same trend. In each single period, decent of the energy is always permitted. Whereas, ascent of that is also permitted with high frequency at the beginning and becomes less likely as the algorithm proceeds. Thus, the CHNN-PSD provides a mechanism for hill climbing by varying the direction of neuron motion. Furthermore, the periodicity is characterized by altering phases of cooling and reheating the stochastic dynamics. Hence, the proposed algorithm provides a means to achieve an effective dynamic or oscillating balance between intensification and diversification during the search. By this way, this mechanism enables the network to search a possible better solution based on the local optimal solution which is obtained in the latest period.

4 CHNN-PSD for Partitional Clustering Problem

The clustering problem, where n objects with D dimensions are clustering into K clusters, can be mapped onto the CHNN-PSD with two dimensions neurons, where it consists of n rows and K columns. Each column is regarded as a group. Thus each group is composed of n neurons.

Let x_{id} ($i = 1,2,\cdots,n, d = 1,2,\cdots,D$) denote the d th features of the object x_i. If i th objects belongs to the k th cluster, the output v_{ik} ($i = 1,2,\cdots,n$, $k = 1,2,\cdots,K$) of the neuron ik is 1. This rule can be described briefly as follows:

$$v_{ik} = \begin{cases} 1 & \text{the } i\text{th object belongs to the } k\text{th cluster} \\ 0 & \text{otherwise} \end{cases}$$

c_k as the center of the k th cluster, is defined in the section 2, then the d th dimension of the k th cluster center is: $c_{kd} = (\sum_{i=1}^{n} v_{ik} x_{id}) / (\sum_{i=1}^{n} v_{ik})$. Therefore, the objective

function of the partitional clustering problem or the energy function of neural network can be defined as:

$$E = \sum_{k=1}^{K}\sum_{i=1}^{n} v_{ik} \left\| x_{id} - c_{kd} \right\|^2 = \sum_{k=1}^{K}\sum_{i=1}^{n} v_{ik} \sum_{d=1}^{D} (x_{id} - c_{kd})^2 \tag{11}$$

By comparing the energy function (11) and the standard energy function (3), we can obtain the connection weights and the biases of the neural network for the clustering problem, with which we can derive the input of the neuron ik for the clustering problem:

$$u_{ik}(t) = -2\alpha(t)\sum_{d=1}^{D}(x_{id} - c_{kd})^2 \tag{12}$$

Note that the neural self-connection is $\omega_{ik,ik} = 0$ and the interconnection between the neurons within the same group is $\omega_{ik,il} = 0$. Therefore $K_{ik,il} = 0$ (ik and il denote the neurons within the same group). Each object belongs to one and only one cluster. Only one neuron is firing ($v_{ik} = 1$) in a column or a group by the winner-takes-all rule. Consequently, according to (7), the input/output function of the i th neuron in the k th group for the clustering problem is given by:

$$v_{ik}(t+1) = \begin{cases} 1 & u_{ik}(t) = \max\limits_{j=1,\cdots n}(u_{jk}(t)) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The CHNN-PSD maybe converge to a so-called degenerate solution, with a partition having less than K non-empty clustering, which also be called empty clustering problem. To avoid this problem, we add a penal term to Eq. (13):

$$u_{ik}(t) = -2\alpha(t)\sum_{d=1}^{D}(x_{id} - c_{kd})^2 + C \cdot e(k) \tag{14}$$

Where the penalty term $e(k) = 1$, if the k th cluster is empty, otherwise, $e(k) = 0$, and C is a reasonable large positive constant.

If the k th cluster is empty, the penalty term can help the neuron ik in the i th group win the competition, which force the object i to be assigned to the k th empty cluster. Thus, the degeneracy (empty cluster) problem can be eliminated effectively by the penal term added.

The following procedure describes the proposed algorithm for the partitional clustering problem:

```
/*Initialization*/
Each object is clustered into one cluster randomly, and
mapped onto the neural network.
for k = 1 to K do
  Evaluate the cluster centers.
end for
/*Main Loop*/
t = 1
```

```
do
for i = 1 to n do
  for k = 1 to K do
  /*Updating Input*/
    Evaluate u_ik(t), according to (14).
  endfor
  /*updating output*/
  for k = 1 to K do
    Evaluate v_ik(t), according to (13).
  endfor
  Update cluster center.
endfor
Keep the best objective, and the best solution.
t = t + 1
until t reaches a given iteration times .
```

5 Simulation Results

The CHNN-PSD is simulated for patitional clustering problem on a notebook Pentium M (1.6GHz, 1.7GHz). In order to help the neuron ik in the i th group win the competition when the k th cluster is empty, parameter C can be set to a relative large positive constant. In the sake of the neuron network having a plenty of choices to increase the energy and convert to a stably state finally, the period T of a single simulated annealing should be set to a relative large positive constant too. Thereby the parameter $\lambda = 50$, $C = 100$ and $T = 300$ were used in the simulation. And the total iteration times are 2400, so the periodic dynamic function has 8 periods.

To show the performance of the CHNN-PSD, three well-known benchmark data sets from the machine learning repository [5], [10] have been considered. These benchmark data sets are fisher's iris, Wisconnsin breast cancer, Ripley's glass and Vowel.

In order to demonstrate the performance of the proposed algorithm, we first compare it with k -means、RS、GA、PSO and DE proposed in [5]. And then, we compare it with CPSO、GA proposed in [6]. The CHNN-PSD is compared with four variations of tabu search, called TS-Clustering, proposed in [7]. The SE and VRC were provided for comparison. Since all the simulations were conducted on benchmark problems, some simulations results were directly adopted from the origin papers [5], [6], [7].

Table 1 shows the mean values and standard errors of SE and VRC over 30 runs on benchmark data sets obtained by k-means, RS, GA, PSO, DE and the proposed algorithm. For Iris data, RS, GA, PSO, DE and the proposed algorithm product the same good result. For cancer data, all the algorithms can get the same good result except the RS. For glass data, the DE and the proposed algorithm can get better result than others. For vowel data, most challenging data set, the proposed algorithm outperforms all other algorithms. All the best results in Table 1 are marked in bold. Note that the proposed algorithm can always acquire the best result in the six approaches on the 4 benchmark data sets with standard error 0.

Table 1. Mean values and standard error over 30 runs on the benchmark data sets

Date set	Criterion	k-means	RS	GA	PSO	DE	CHNN-PSD
Iris	SE	8950.18 (449.79)	**7885.14 (0)**	**7885.14 (0)**	**7885.14 (0)**	**7885.14 (0)**	**7885.14 (0)**
	VRC	-	**561.63 (0)**	**561.63 (0)**	**561.63 (0)**	**561.63 (0)**	**561.63 (0)**
Cancer	SE	**19323 (0)**	19361 (3.3296)	**19323 (0)**	**19324 (0.3944)**	**19323 (0)**	**19323 (0)**
	VRC	-	1022.28 (0.3555)	**1026.26 (0)**	**1026.26 (0.0002)**	**1026.26 (0)**	**1026.26 (0)**
Glass	SE	366.98 (6.59)	393.21 (2.223)	341.09 (1.5999)	339.04 (0.9505)	**336.06 (0)**	**336.06 (0)**
	VRC	-	101.05 (0.9564)	121.94 (0.7799)	122.74 (0.578)	**124.62 (0)**	**124.62 (0)**
Vowel	SE	31687462 (229932)	34898130 (183388)	30943106 (13606)	30734068 (6467.23)	30690785 (1816.64)	**30686238 (0)**
	VRC	-	1278.7 (7.14)	1450.45 (0.9677)	1463.33 (0.3809)	1465.55 (0.1156)	**1465.88 (0)**

The worse, average and best result over 10 runs obtained by CPSO with heuristic initialization and the proposed algorithm are shown in Table 2. For Iris and cancer data, the two algorithms both can get the same good results. While the proposed algorithm outperforms the CPSO on the glass and vowel data, especially in terms of mean and worse result objective.

The minimum, average and standard error results over 10 runs provided by KM-DHB, DHB-DHB, DHF-DHB, as three variants of TS-Clustering , and the CHNN-PSD, on the vowel data, are shown in Table 3. From the table, we can observe that all the algorithms in the table can get the same good minimum value and the proposed algorithm performs much better than other approaches for the average and standard error values.

Table 4 shows average computation time (second) or the number of fitness evaluation function of the algorithms mentioned above on benchmark data From the table, we can see that the proposed algorithm is faster than RS, GA, PSO, DE and CPSO in their own environments, while slower than k-mean because the proposed algorithm has mechanism to escape from local minima.

Table 2. Experimental results over 10 runs on the benchmark data sets (CPSO with heuristic initialization)

Data set	Criterion	CPSO			CHNN-PSD		
		Worse	Average	Best	Worse	Average	Best
Iris	SE	7885.14	**7885.14**	**7885.14**	7885.14	**7885.14**	**7885.14**
	VRC	561.63	**561.63**	**561.63**	561.63	**561.63**	**561.63**
Cancer	SE	19323	**19323**	**19323**	19323	**19323**	**19323**
	VRC	1026.26	**1026.26**	**1026.26**	1026.26	**1026.26**	**1026.26**
Glass	SE	375.79	336.17	**336.06**	336.27	**336.10**	**336.06**
	VRC	123.62	124.44	**124.62**	124.51	**124.56**	**124.62**
Vowel	SE	30712302	30691704	30689508	30686238	**30686238**	**30686238**
	VRC	1465.47	1465.65	**1465.88**	1465.88	**1465.88**	**1465.88**

Table 3. Minimum, average and standard error values over 10 runs on the Vowel data sets

Data set	Criterion		KM-DHB	DHB-DHB	DHF-DHB	CHNN-PSD
Vowel	SE	Avg.	30687595.43	30686426.39	30686527.05	**30686238.38**
		Min.	**30686238.38**	**30686238.38**	**30686238.38**	**30686238.38**
		SD.	935.68	376.03	441.77	**0.61**

Table 4. Average computation time (second) on benchmark data sets

	k-mean	RS, GA, PSO, DE	CPSO and GA	CHNN-PSD
Number of Fitness Evaluation	-	100,00 times	250,00 times	-
Iris	0.03	27	-	0.88
Vowel	0.19	270(RS:250)	-	9.84

From all simulation results, we can conclude that the proposed algorithm can find better or comparable solutions within relatively less computation time than previous approaches.

6 Conclusions

This paper presents a novel competitive Hopfield neural network with periodic stochastic dynamics to solve the partitional network. The periodic stochastic dynamics is introduced to help the neural network escape from the local minima and search a possible better solution based on the best solution which is obtained in the latest period. The proposed algorithm is evaluated through a series of well-known benchmark data sets. The simulation result reveals that the proposed algorithm can get better or comparable results, with relatively less computation time than previous approaches, such as k-means, genetic algorithm, particle swarm optimization, differential evolution, combinatorial particle swarm optimization and TS-clustering.

Acknowledgments. The research work described in this paper was supported by the Guangdong Provincial Natural Science Foundation of China (07300630), the Specialized Research Fund for the Doctoral Program of Higher Education (20070558052), and the Scientific Research Foundation for the Returned Overseas Chinese Scholars State Education Ministry.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
2. Forgy, E.: Cluster analysis of multivariate data: efficiency versus interpretability of classification. Biometrics 21, 768–769 (1965)

3. Bhuyan, J.N., Raghavan, V.V., Venkatesh, K.E.: Genetic Algorithm for Clustering with an Ordered Representation. In: Proc. Fourth Int'l Conf. Genetic Algorithms, pp. 408–415 (1991)
4. Krishna, K., Murty, M.N.: Genetic k-means algorithm. IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetic 29(3), 433–439 (1999)
5. Paterlini, S., Krink, T.: Differential evolution and particle swarm optimization in partitional clustering. Computational Statistics & Data Analysis 50, 1220–1247 (2006)
6. Jarboui, B., Cheikh, M., Siarry, P., Rebai, A.: Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. Applied Mathematics and Computation 192(2), 337–345 (2007)
7. Liu, Y.G., Zhang, Y., Wu, H., Mao, Y., Chen, K.F.: A tabu search approach for the minimum sum-of-squares clustering problem. Information Sciences 178, 2680–2704 (2008)
8. Hopfield, J.J., Tank, D.W.: "neural" computation of decisions in optimization problems. Bio. Cybern. (52), 142–152 (1985)
9. Galán-Marín, G., Mérida-Casermeiro, E., Muñoz-Pérez, J.: Modelling competitive Hopfield networks for the maximum clique problem. Computer & Operations Research 30(4), 603–624 (2003)
10. Merz, C., Murphy, P., Aha, D.: UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine (1997), `http://mlearn.ics.uci.edu/MLRepository.html`

Modeling Hysteresis in Piezo Actuator Based on Neural Networks*

Xuefeng Yang, Wei Li, Yuqiao Wang, and Guo Ye

College of Mechatronics Engineering, China University of Mining and Technology, Xuzhou 221008, China
hopeasy@163.com

Abstract. Hysteresis and nonlinearity of piezo actuator are the major factors affecting the motion accuracy in controlling a micro system. The prediction accuracy of classical Preisach model could be improved only by mass experiments for measuring hysteresis. A model based on BP neural networks was proposed to improve the prediction accuracy. Because the stoke of piezo actuator have relation to historical extrema inputs, the input of the model are current exciting voltage, historical voltage at nearest turning point and its corresponding stroke and the output is piezo actuator's stroke. Results of simulation and experiments show that the proposed hysteresis model can exactly describe and predict the hysteresis of piezo actuator compare with traditional bilinear interpolation and has the better generalization ability.

Keywords: hysteresis, BP neural networks, Preisach model.

1 Introduction

With the rapid development of precision machining and nanotechnology, precision positioning technology with the accuracy of submicrometer even nanometer is increasingly needed in the fields of precision machining, biological engineering and microelectronics. There is an widely usage of piezoelectric actuators, because they have compact structure, high resolution, fast response, high stiffness, no backlash and no friction. A piezo actuator, however, exhibits hysteresis nonlinearity, which severely limit the scope of its application. Therefore, more and more people concern about how to describe the behavior of hysteresis nonlinearity correctly and eliminate its affects [1][2].

The behavior of hysteresis nonlinearity can result in the degradation of system performance and even lead to instability [3]. In order to eliminate the effect of nonlinear hysteresis of a piezo actuator to the positioning accuracy, many scholars at home and abroad proposed lots of methods. Among them, P. Ge [4] and Song D [5] adapted the Preisach model[6] based on the phenomenon of piezo actuators to describe the hysteresis. This method merely considered the input and output relations of piezo actuator and omitted the microstructure of piezo materials, has received much attention recently for its excellent properties and easy implementation compare to other meth-

* This work is partially supported by Graduate Education Innovation Project, Jiangsu province
Corresponding author. Tel: +86-13395226900; E-mail address: hopeasy@163.com

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 290–296, 2008.
© Springer-Verlag Berlin Heidelberg 2008

ods. The disadvantages of the Preisach model are the dependence on experimental samples and employing bilinear interpolation.

Neural networks have been widely used because of its high precision approximation capabilities and a strong ability to fault-tolerant in the identification [7][8]. Neural networks, applied to hysteresis molding, are mainly used in mathematics formula improvement, have the same shortcoming with Preisach model. This paper study on the hysteresis of piezo actuator based on neural networks in the frame of Preisach model and a new model is established. Results of simulation and experiments show that the proposed hysteresis model is sample and accuracy.

2 Summary of Preisach Model

Piezo actuator is composed by stacking many single PZT layer like piezo-capacitor. When this piezo-capacitor is exposed to electric fields, the deformation is created. Piezo stack actuators make use of the increase of the ceramic thickness in direction of the applied electrical field to produce motions and/or force. Because of the electricity physical property of piezocrystal, charging and discharging lead to the hysteresis nonlinearity phenomenon. Voltages and stokes trace out a series of hysteresis loop was formed at voltage-stroke plane.

The Classic Preisach Model is constructed by summarize a mass of Preisach-plane-weighing-operators $\gamma_{\alpha\beta}$. Each of operators has the basic characteristics of hysteresis loop. Figure 1 shows the input and output of the operator [9].

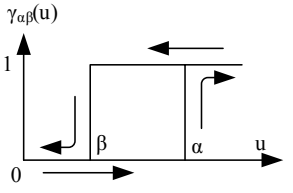


Fig. 1. Hysteresis operator

Due to the direction of voltage $u(t)$ applied must be along with the polarized direction of piezo actuator, that is, the value of input voltage $u(t)$ and the output stroke $x(t)$ is positive at anytime. Therefore, the hysteresis loop of the hysteresis operator $\gamma_{\alpha\beta}$ is either 0 or 1. A threshold (α, β) decides rising edge and falling edge of operator, α and β are the input values where up and down switching occur, respectively ($\alpha \geq \beta$). Then, the Preisach model is expressed as follow:

$$x(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta} u(t) d\alpha d\beta \tag{1}$$

Where, $\mu(\alpha,\beta)$ is weight function. Hysteresis operator $\gamma_{\alpha\beta}$ has simulated the general hysteresis phenomenon, and the integration form concerned the influence of historical inputs

The Preisach model has two important properties, i.e. the wiping-out property and the minor-loop congruence property. The wiping-out property is that the input extremum voltage, at some point, large than the historical input maximum voltage or small than the minimum voltage, the new extremum voltage will 'wipe out' the historical extremum value and the corresponding trajectory. The wiped out value do not affect the output stroke after this time. A minor-loop is the closed loop in the input-output plane when the input cycles between two extrema. The minor-loop congruence property states that all the comparable minor loops are identical in shapes.

The Preisach model can describe the hysteresis nonlinearity of piezo actuator easily. However, in formula (1), there is a double integral, which is difficult for computing. Meanwhile, the weighing function $\mu(\alpha, \beta)$ should calculate the second order derivative of the experimental data, which is not propitious for computer realization. According to reference [4], the following two equations provide the explicit numerical forms of Preisach model.

When $u(t)$ is ascending,

$$x(t) = \sum_{k=1}^{n-1} [x(\alpha_k, \beta_{k-1}) - x(\alpha_k, \beta_k)] + x(\mu(t), \beta_{n-1}) \tag{2}$$

when $u(t)$ is descending,

$$x(t) = \sum_{k=1}^{n-1} [x(\alpha_k, \beta_{k-1}) - x(\alpha_k, \beta_k)] + [x(\alpha_n, \beta_{n-1}) - x(\alpha_n, \mu(t))] \tag{3}$$

Where $x(\alpha_k,\beta_{k-1})$, $x(\alpha_k,\beta_k)$, are past input extrema which affect the output.

3 Modeling Hysteresis Based on Neural Networks

3.1 Description of Hysteresis

Figure 2 shows the generation of multi-loop hysteresis. (1) Input voltage $u(t)$ rise form 0 to $u(b)$ and the corresponding change of stroke is from 0 to $f(b)$; (2) The voltage $u(t)$ fall from $u(b)$ to $u(c)$ and the stroke is from $f(b)$ to $f(c)$ correspondingly. According to Preisach model, the minimum a at curve ab affect the output stroke of ab, and the maximum b at curve bc affect the output stroke of bc. (3) $u(t)$ rise form $u(c)$ to $u(d)$ and the stroke is from $f(c)$ to $f(d)$. b and d are coincidence point and constitute a closed loop according to minor-loop congruence property. c is the minimum at cd and affect the its output stroke. (4) The input voltage is rising to $u(e)$ and the corresponding stroke is coincidence with main loop at $f(d)$ and rise to $f(e)$. Then the influencing factor of output stroke is a rather than e according to wiping-out property. (5) Stroke fall from $f(e)$ to 0 when the voltage is from $u(e)$ to 0 and the affect point change to e. The extreme point affect the output stroke vary along with $u(t)$ and generate the hysteresis loop.

In voltage-stroke plane, the extreme points that affect the output stroke are called turning points. Therefore, the turning point of curve ab and ade is a, bc is b, cd is c, ea is e.

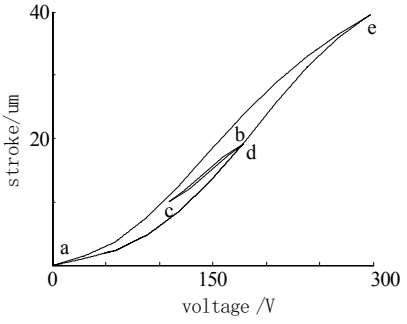


Fig. 2. Generation of multi-loop hysteresis

In order to simplify the measurement, we firstly divide the voltage range from 0 to saturation α_s into n equal parts in practical applications. Traversing all the up and down trajectory passing each equidistributed point, we obtain a Preisach function table. If the voltage is at the equidistributed points, $x(\alpha, \beta)$ can be obtained by lookup table, otherwise, $x(\alpha, \beta)$ can be calculated by bilinear interpolation that make use of four nearest value in Preisach function table. Therefore, for any sequence of the control voltage, the output stroke of a piezo actuator can be predicted by Preisach model [10].

According the method above, we must divide the input voltage equally and excite piezo actuator with special sequence. In addition, the voltage accurate at equidistributed point will affect the precision of stroke. Interpolation will bring error when the voltage is not among the equidistributed points. The equidistributed point should be large if we want to improve the prediction accuracy, this greatly increase the complexity of experiment process. This paper employs the neural networks to model the piezo actuator hysteresis. The input is random voltage instead of special sequence and record the output stroke, which eliminate the voltage error at equidistributed point. There we make use of the neural network's ability of approximate any continuous function with any degree of accuracy to predict the output stroke.

3.2 Modeling Hysteresis Based on Neural Network

Neural networks are interconnected by mass of processing unit (neurons). An artificial neural network is proposed based on the study of modern neuroscience to simulate the brain function. BP neural network, has strong nonlinear mapping ability and better generalization ability in those neural networks, are widely used many fields [11].

Current output stroke of piezo actuator has relation to current input voltage and the recent historical turning point. Then the input sequence voltage $[u(t),u_m(t),x(u_m(t))]$ was expanded to a vector group. Where, $u(t)$ is input voltage at time t, $u_m(t)$ is input voltage at nearest turning point and $x(u_m(t))$ is its corresponding stroke.

Therefore, a model based on BP neural networks is build, the input of the model are current exciting voltage, historical voltage at nearest turning point and its corresponding stroke and the output is piezo actuator's stroke. Topologic structure is shown in figure 3, the network is constituted by input layer, output layer and two

hidden layer. Each hidden layer has five neurons, and the transfer function is sigmoid function. The train method of the BP neural networks is Bayesian normalization [12].

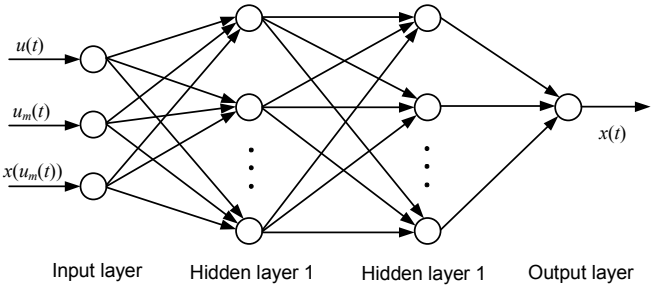


Fig. 3. Topologic structure of BP neural network

4 Simulation and Experiment

The WTDS0808055 type piezo actuator made by CETC NO.26 RESEARCH INSTITUTE is adapted to verify the hysteresis model base on BP neural networks. The driving voltage of the piezo actuator ranges from 0 to 300V and the maximum stroke is 39.95um.

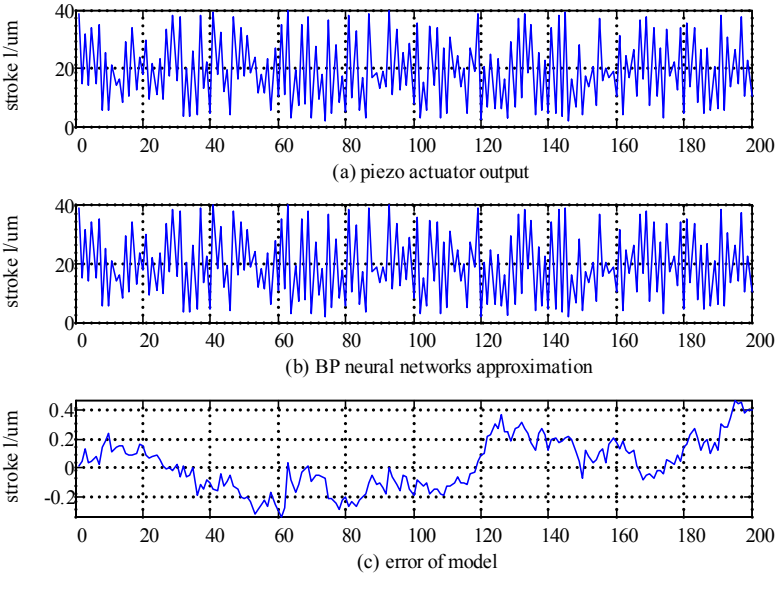


Fig. 4. Approximation of the model

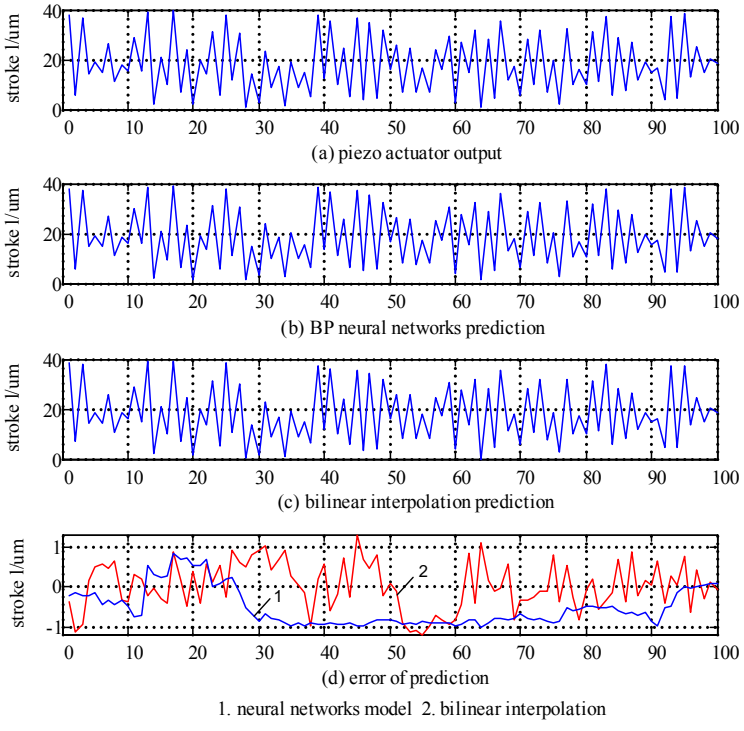


Fig. 5. Prediction error of the model

Input voltage sequence is random signal vary from 0 to 300V and record the corresponding output stroke. Meanwhile, the turning point was calculated. Those are training data of BP neural networks shown in figure 3. Figure 4 shows the approximation of the model, the maximum error is 0.4673 and the standard deviation is 0.1748.

In order to verify the generalization ability and validation of model, a new 100 random voltage is used to excite the piezo actuator and the output stroke is recorded. Meanwhile, the prediction of neural networks model compare with the bilinear interpolation's prediction. Where, the equidistributed point n is 20, i.e. the Preisach function table should be fill two hundred data in bilinear interpolation experiment. Figure 5 shows the piezo actuator's output and the model's prediction. The maximum error is 0.8522 and the standard deviation is 0.4736 with neural networks while the bilinear interpolation is 1.3034 and 0.5873, separately. Results of experiments show that the BP neural networks model has some advantages in bringing down the piezo actuator's output hysteresis error.

5 Conclusions

The prediction accuracy of classical Preisach model could be improved only by mass experiments for measuring hysteresis and the stroke not among the equidistributed points is calculated by interpolation. That is why the classical Preisach model has low

accuracy and is difficult to apply to online control. This paper establish a neural networks model based on classical Preisach model and study on the identification and prediction of piezo actuator's hysteresis. Finally, the output of model is verified by the stroke of piezo actuator and the prediction of bilinear interpolation. Results of experiments show that the BP neural networks model can improve the accuracy of prediction effectively and overcome the shortcoming of classical Preisach model which improve the prediction accuracy only by enlarge the number of equidistributed point. What's more, the model can exactly describe the hysteresis of piezo actuator and has the better generalization ability.

References

1. Yu, Y., Nagi, N., Rao, D.: Preisach modeling of hysteresis for piezoceramic actuator system. Mechanism Machine Theory (37), 49–59 (2002)
2. Chih, L.H., Chau, J., Ye, H.C.: Piezomechanics using intelligent variable-structure control. IEEE Transactions on Industrial Electronics 48(1), 47–59 (2001)
3. Tao, G., Kolotovic, P.V.: Adaptive control of plants with unknown hysteresis. IEEE trans. Automatic Control 40(2), 200–212 (1995)
4. Ge, P., Jouaneh, M.: Tracking control of a piezoceramic actuator. IEEE Transaction on Control System Technology 4(3), 209–216 (1996)
5. Song, D., James, C., Modeling, L.: of piezo actuator's nonlinear and frequency dependent dynamics. Mechatronics 9(4), 393–410 (1999)
6. Mayergoyz, I.D., Friedman, G.: Generalizd Preisach model of hysteresis. IEEE Transactions on Magnetics 24(l), 212–217 (1988)
7. Adly, A.A., Abd-El-Hafiz, S.K.: Using Neural Networks in the Identification of Preisach-Type Hysteresis Models. IEEE Transactions on Magnetics 34(3), 629–635 (1998)
8. Serpico, C., Visone, C.: Magnetic Hysteresis Modeling via Feed-Forward Neural Networks. IEEE Transactions on Magnetics 34(3), 623–628 (1998)
9. Ge, P., Jouaneh, M.: Modeling hysteresis in piezoceramic actuators. Precision Engineering (17), 211–221 (1995)
10. Hen, G.Q., Wei, Y.D.: Study on Nonlinear Model of Piezoelectric Actuator and Accurate Positioning Control Strategy. In: Proceedings of the 6th World Congress on Intellignet Control and Automation, pp. 8356–8360 (2006)
11. Sun, Y.J., Zhang, S., Miao, C.X., et al.: Improved BP Neural Network for Transformer Fault Diagnosis. Journal of China University of Mining and Technology 17(1), 138–142 (2007)
12. Lauret, P., Fock, E., Randrianarivony, R.N.: Bayesian neural network approach to short time load forecasting. Energy Conversion and Management 34(4), 1156–1166 (2008)

Modeling of Grey Neural Network and Its Applications

Jingling Yuan, Luo Zhong, Xiaoyan Li, and Jie Li

Computer Science and Technology School, Wuhan University of Technology,
430070, Wuhan, China
yjl@whut.edu.cn

Abstract. Grey neural network is an innovative intelligent computing approach combing grey system model and neural network, which makes full use of the similarities and complementarity between grey system model and neural network to settle the disadvantage of applying Grey model and Neural Network separately. Some optimization algorithms such as genetic algorithm are also employed to modeling and optimizaion of grey neural network. Many typical grey neural work models such as GNNM(1,1),GRBF,DGRBF, GA-GRBF and so on are proposed and applied in this paper. A lot of comparative experimental results show that grey neural network models are capable of predicting a small sample of data accurately, easily and conveniently. The key technologies, research hotspots, difficulties and further development of grey neural network are discussed in this paper.

1 Introduction

It is a critical problem nowadays to study on information, especially the expression and process of uncertain information. The first uncertainty is randomicity which is studied deeply, forms probability, statistic, and information theory and is applied in a wide range. We all know that the fuzzy mathematics built by L.A.Zadeh is the tool to process fuzzy information. Professor Deng Ju-Long in the early 80th established Grey system theory in which the research object is uncertain system with small sample and poor data information with the characteristic of having some known and some unknown information. This system is to process the known information with some methods such as renewing, developing and analyzing to get valuable and regular information for correct description and effective controlling. Professor Liu Kaidi enhanced grey mathematic method by some theories such as grey set and grey number[4]. Randomicity, fuzzy, and grey are the uncertainty acknowledged by scholars over the world and are applied in industry and research.

The grey model has some advantages such as required small samples, need not to consider the distribution law and trend, easy to modeling and computing. But it has some disadvantages like lack of the ability of self-learning, self-organize, and self-adaptive, weak in processing nonlinear information, without taking the feedback adjustment of error in the process of modeling of feedback correction into consideration, and low accuracy. And as we all know that the neural network has strong capacity of processing information. But it has some disadvantages such as without taking some information with certainty as instructing information as a black box, need a large

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 297–305, 2008.
© Springer-Verlag Berlin Heidelberg 2008

amount of learning sample, and ignore some information characteristics of some information with uncertainty and the applications of some information with certainty. Therefore, the grey models' characteristics such as poor information, modeling easily, and weak in nonlinear processing and neural networks' characteristics such as large amount of sample, nonlinear processing ability, and learning ability compliment and enhance organic with each other. In the real world, the problem, which cannot solve just by neural network, with incomplete information and lack of learning samples is wide spread. Therefore we take the advantages of both to build high-performance neural network model[9].

2 Fusion of Neural Network and Grey System

Because the way of presenting information of neural network and grey model has some in common, they can be fused. The neural network can proximate certain value in certain precision, but it fluctuates for the existence of some error. We can use u(x) to represent, where x is to represent the expect output. Moreover, according to the definition of grey number in the grey system theory, we can get that the output of BP network is a grey number. Therefore, the neural network itself has some grey content. So, we can use grey system theory to analyze neural network, and can research grey system through neural network. Research of Fusion of Grey system and Neural network

(1) Simple combination of neural network and grey system
In complex system, we can use grey system method and neural network at the same time, which is to use grey system method to process the part with obvious grey characteristic and without distributed parallel computation, and to use neural network to process the part without grey characteristics and a kind of like black box. In this method, neural network and grey system have indirect connection.

(2) Series Combination
In this method, grey model and neural network combine in series way, which is the output of one of them is the input of the other of them, to predict fault-tolerant analysis of complicated system.

(3) Enhance grey system with neural network
Grey system modeling is aim at constructing differential equation. Because the empty set in information time zone (where has no information time zone), we can construct the approximate and partly uncertain differential equation through the conditions of approximate differential equation. But, practically, it is very hard to use grey differential equation directly. Therefore, we need whitening the grey differential equation. We consider parameters of grey differential equation when we construct BP network, and make them in the network. The BP network is trained by the sample abstracted from grey system. When BP network is convergence, we can abstract whitening parameters of grey differential equation to get certain accuracy to obtain the continuous modeling.

(4) Grey system assists in constructing neural network
For the grey system's structure of information including certain information and uncertain information, when we use neural network to solve grey system, the certain

information can be used to construct neural network and improve the learning method of neural network.

(5) Completely fusion of neural network and grey system

The complete fusion of neural network and grey system is grey neural network which apply difference methods to different neural network.

①Hopfield network：The grey information is used to construct grey neural, and the certain information is used to construct general neural. And the network construction method and energy function are improved according to the characteristics of grey system.

②BP network：Based on general BP network, according to the grey system dynamics and its certain information, we add a grey level before to process the input information grey, and add a whitening level after to process the output information of grey to get certain result [13,14].

3 Grey Neural Computing Model

3.1 Modeling Method of Grey Neural Network

The fusion of grey system and neural network has 3 categories which is series type, parallel type and embed type. Series type neural network is to process information by grey model and neural network separately, and then combine all the result. Parallel type is to combine the result of several grey computing modeling by neural network. Embedded type is to add a grey level and whitening level to the input and output separately. The grey neural network whitening grey differential equation is an example of the embedded type. Therefore, we study on different grey model and different neural network, and combine by different way to satisfy different requirement. BP network model is an example of the embedded fusion.

Based on general neural network, according to the grey system dynamics and its certain information, we add a grey level before to process the input information grey, and add a whitening level after to process the output information of grey. We research on the structure of neural network, the dynamics of grey system, and the certain information's influence to the number of hidden node, and to find some rule in it. We mainly build 6 grey BP neural network model, that is GNNM(1, 1), GNNM(2, 1), GNNM（1，2），GNNM（1，3），GNNM（1，4）and DNNM（1，4）which define the processing and computation method of 1-dimension or n-dimension grey problem. First two concentrate on modeling of 1-dimsion grey problem and the others concentrate on modeling of 2-dimension, 3-dimension, and 4-dimension grey problem. The object which these model serve have different main parameters[10-16].

3.2 Application of Grey Neural Network Prediction Model

Nowadays, Grey neural network is widely used in prediction. Chen Shuyan use series type, parallel type and embed type grey neural network model to predict the traffic flow of jing-shi express way, and the accuracy and the result of prediction is pretty good, better then signal prediction model. The result of the experiment indicates the grey neural network can improve the accuracy of prediction and it can be use to

predict traffic flow. Gao Wei, Fan Wei propose the method of combine grey GM(1,1) and series type neural network to build prediction model of grey neural network to predict displacement of landslide. The author used the grey neural network to find the relationship of the prediction value and measured value to modify prediction value of grey system. This prediction model has high accuracy, need fewer sample, effective, and has great future in landslide prediction. Ruan Ping, Gu Yihua proposed combine prediction model of multi-parameter neural network to predict power load and to build optimized combination. The result of their computation indicates the method proposed take power load and several factors into consideration to improve the prediction accuracy. Some other scholars combine grey prediction and neural network to propose a new method to predict macro economic index. Zhong Luo□Yuan jingling proposed grey RBF prediction model to predict wind's response time-history, and obtain very good accuracy[2,3].

3.2.1 GRBF Prediction Model[2]

In many engineering projects, because of the limitations on the tools of measurement and methods used for computation and analysis, data gathering is always a tough task. In some previous models, such as GM(0, N), a linear static model, could not accurately predict when connections between dependent and independent variables involving nonlinear relationships or the relationship is unknown; the RBF, which also has an obvious drawback that the precision and stability of the model are highly dependent on the volume of sample data, so when sample data are relatively small, the results always are not acceptable in precision and thus has little value on this issue. In an attempt to solve this problem, a new GRBF (Grey Radial Basis Function) prediction model, which combines the character of accurate prediction with a small volume of sample data of GM (0, N) and advantage of rapid convergent speed for nonlinear functions of RBF Neural Networks, is proposed. Therefore, we should do 1-AGO (once Accumulated Generating Operation) to the samples in the first step of building GM(0, N), so as to increase the linear characters and reduce the randomicity of original samples. Then train the accumulated samples with RBF, and the resulting function is our wanted GM(0, N). Finally use the GM(0, N) to predict unknown values, and perform IAGO (Inverse Accumulated Generating Operation) to generate the final predicted values. The whole process described above is then the GRBF model introduced in this paper. The model fully takes advantages of RBF and GM(0, N), and rids off the drawbacks of them, achieves the goal of effective and accurate. The details are showed in reference 2.

The new combination prediction model has been applied in the Intelligent Health Monitoring System for Large-Scale Roof Lattice Structure of Shenzhen Civil Center successfully and proved to be an easy, convenient and accurate method. The following figure shows the performance of GRBF. From the figure, one can see that the predicted values from RBF have the largest relative error for the reason that it requires large volume of samples; predicted values from GM (0, N) are then fairly close to the actual values, with average relative error of 6.7809%; finally the predicted values from GRBF have the highest accuracy, with some points almost overlapped the actual values and the average relative error is 3.4628%. Therefore, we can conclude that the new GRBF Prediction works effectively and has the highest prediction accuracy of 96.5372%.

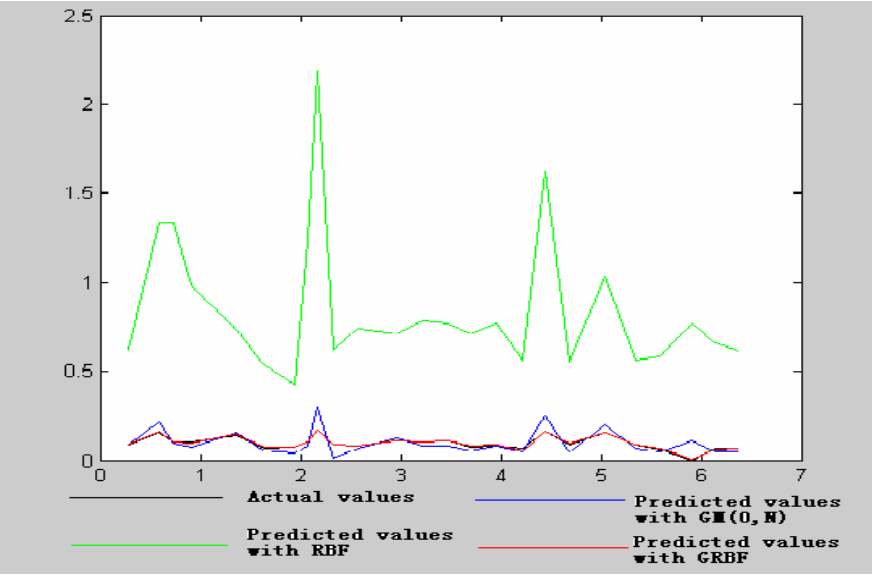


Fig. 1. Prediction performances among GRBF, GM(0, N), RBF

3.2.2 DGRBF Prediction Model[1,3]

The dynamic Gray Radial Basis Function (DGRBF) prediction model is the improvement of traditional dynamic grey prediction model GM (1,1). It gives the dynamic algorithm of acquiring optimized initial conditions and identifying parameters like metabolism, and then the model combines the characters of RBF Neural Networks, and therefore has the ability of dynamic prediction on small volume of samples. The DGRBF prediction model has been also applied in the Intelligent Health Monitoring System for Large-Scale Roof Lattice Structure of Shenzhen Civil Center successfully, and the experiment results demonstrate that model and its intelligent optimizing algorithm are capable of predicting in a long term and the desired data could be acquired accurately, easily and conveniently. The structure of DGRBF showed in figure 2, RBF is just like the compensator of errors. The details showed in reference 3.

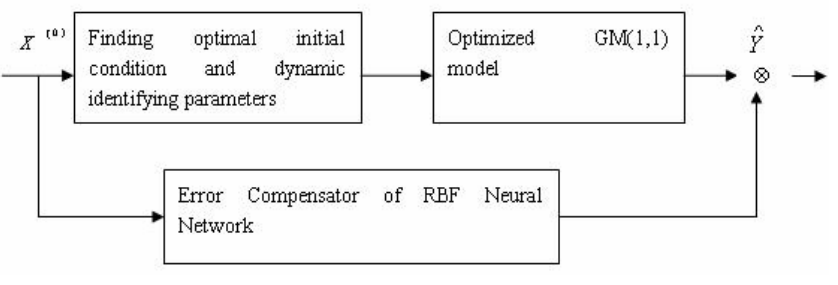


Fig. 2. The structure of DGRBF Model

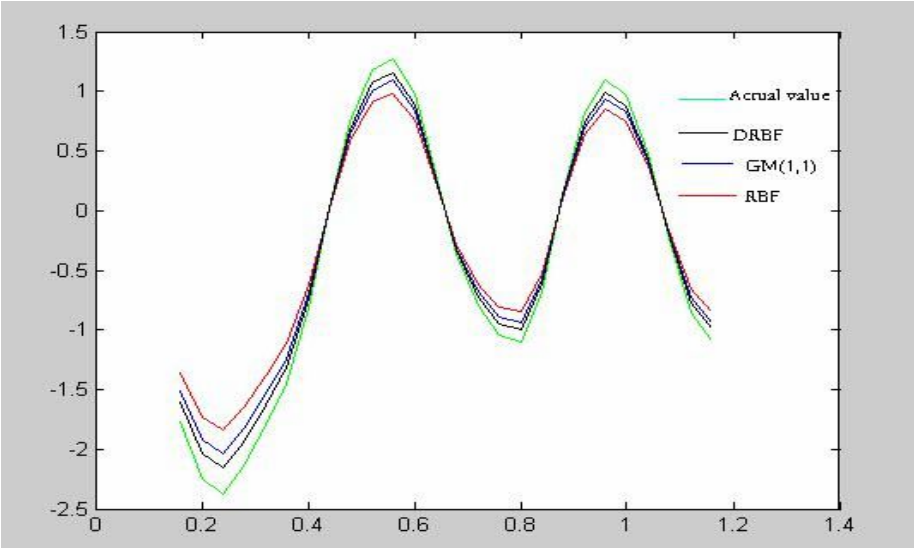


Fig. 3. Prediction comparison among RBF, GM(1,1), DGRBF and actual values

The following figure shows the DGRBF with a real application and compare it with RBF and GM(1,1). It shows the curves of DGRBF, RBF, traditional GM(1,1) and the actual values. Obviously, the black curve is the closest to the green one, which also means that the predicted values of DGRBF, whose average error is 3.58%, are the closest to the actual values. And the RBF has the largest error due to the reason that the volume of samples is too small.

3.3 Whitening Method of Neural Network with Grey System[13-14]

Using Neural network to abstract grey model parameter and whitening parameter of grey differential equation of grey system to make up the disadvantages of grey system in the process of whitening. According to the structure of grey differential equation, characteristics of parameters and the certain information of grey problem, the grey neural network is constructed. Research on how to use the parameters of grey differential equation is one part of grey neural network, which is to abstract training sample from the known data and some data after modification to train neural network. However, grey neural network in the stable state and can abstract the parameter after whitening to get grey differential equation after whitening which is a certain differential equation. We find the relationship of parameter in grey differential equation with the weight and threshold of neural network by mapping, and can get the weight and threshold of neural network after training to whitening parameters of grey differential equation, and study on structure and relative algorithm of the neural network which apply whitening on grey differential equation. Zhong Luo whiten the parameters of grey differential equation through BP neural network and obtain pretty good result. The details showed in reference 13,14.

4 The Optimization of Grey Neural Computation Model

The optimization of grey neural computation model is a difficulty of the research of grey neural computation model, which has not been researched deeply. Nowadays research is limited to the optimization of neural network itself, without any optimization of grey neural computation model. Therefore, we consider we should research from three aspects as follows:

(1) Analyze the key parameters of neural network and grey system deeply. On one hand, we introduce intelligent algorithms (such as genetic algorithm, simulated annealing algorithm and so on) to optimize the center parameter of neural network, and train the weight to get the undetermined parameters; on the other hand, we optimize the parameter of grey system by proper algorithms. Then we can combine neural network and grey system model to improve the accuracy and practicability of grey neural network model.

For example, when combining grey system with RBF neural network local optimization and convergence problems still exist, so genetic algorithm is introduced to assist the modeling of grey neural network. Firstly, genetic algorithm is employed to solve the parameters of improved GM(1,1) with Lagrange's Mean Value Theorem, and then RBF neural network is parallel connected to compensate errors. A new dynamic prediction model integrating genetic algorithm and grey RBF, for short

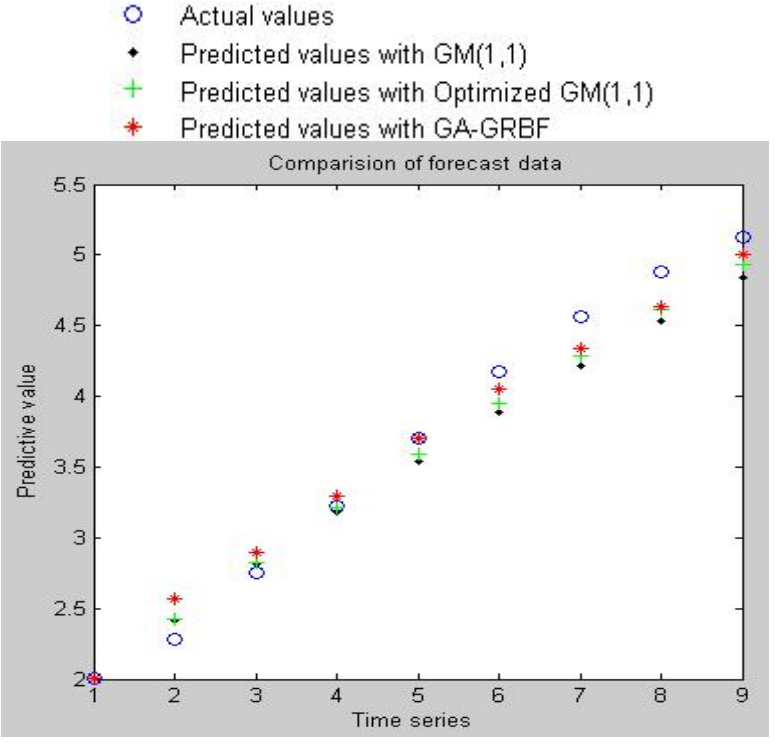


Fig. 4. Comparison between Predictive values of Different Models and Actual values

GA-GRBF is proposed. This new model with preferable structure and parameters is applied to simulation and analysis of time-displacement data of wind response. The comparative experiment results show that this model is capable of predicting a small sample of data accurately, easily and conveniently. The figure 4 mainly compare traditional GM(1,1), Genetic Algorithm Optimized Metabolic GM(1,1)and GA-GRBF. Obviously, GA-GRBF is also capable of predicting the latter values.

(2) We research the neural network technology and characteristics of grey system deeply, mainly focus on the network structure of grey neural computation model (such as introducing the structure risk function of support vector machine, optimizing grey neural network predication model), to raise the performance of grey neural computation model (including computation speed, computation accuracy and so on).

(3)Since we use grey nerve cell、characteristics of grey system and definite information of grey problems, the learning method of grey neural computation is different from that of the traditional neural network, and we should research several learning methods adapting to different grey neural computation.

5 Conclusion

Grey neural computation model is a kind of computation model based on analysis of characterics of grey system and neural network, combining two technologies organically. It makes full use of the existing similarity of these two technologies on information performance and the existing complementarity on model characteristics, and it can make up the disadvantage of soly using grey model or neural network to solve problems. Meanwhile, using neural network to whiten the parameter of grey differencial equations can make up the disadvantage of grey system in whitening parameters. Grey neural network predication model has a pretty extensive application, whose result is quite perfect. This model provides novel thought and effective method for the expression of undetermined information and complicated problems modeling. However, the optimization of grey neural computation model has not been researched deeply, and the analysis method of grey neural computaion performance is expected.

References

1. Yuan, J., Zhong, L., Tong, Q., et al.: The Modeling of Metabolic GM(1,1) Prediction Model and Its Application. In: Progress in Intelligence Computation and Application, ISICA 2005, pp. 718–722 (2005)
2. Yuan, J., Zhong, L., Jiang, Q.: A Study on Grey RBF Prediction Model. In: Yeung, D.S., Liu, Z.-Q., Wang, X.-Z., Yan, H. (eds.) ICMLC 2005. LNCS, vol. 3930, pp. 4140–4143. Springer, Heidelberg (2006)
3. Yuan, J., Zhong, L.: The Dynamic Grey Radial Basis Function Prediction Model and its Applications. In: Proceeding of IEEE ICICIC 2006, pp. 582–585 (2006)
4. Liu, K.: Mathematic process and application of uncertain information. Science Press house, Beijing (1999)
5. Gao, W., Feng, X.: Study on displacement prediction of landslide based on grey system and evolutionary neural network. Rock and Soil Mechanics 25(5), 514–517 (2004)

6. Ruan, P., Lei, Z., Wang, H.: Long-term power load prediction based on grey system and neural network. Computer applications 24(6), 285–286 (2004)
7. Zhong, L., Rao, W.: Neural network and its fusion technology. Science Press house, Beijing (2007)
8. Lu, H., Zhang, Y., Yu, Q., Li, H.: Grey forecasting model based on evolving neural network. Journal of PLA University of Science and Technology(Natural Edition) 7(5), 437–441 (2006)
9. Chen, S., Chen, J.: Application of a novel model to Traffic Flow Prediction. Journal of highway and transportation research and development 21(2), 80–83 (2004)
10. Zhong, L., Liu, L., Zou, C., Yuan, J.: The Application of Neural Network in Lifetime Prediction of Concrete. Journal of Wuhan University of Technology 17(1), 79–81 (2002)
11. Zhong, L., Yuan, J., Xia, H., et al.: A Study on Gray Neural Network Modeling. In: Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, China, pp. 2021–2023 (2002)
12. Lv, H., Zhong, L., Xia, H.: Exploring in Fusion Technology of Grey system and neural networks. Micro-Computer development 10(3), 3–5 (2000)
13. Zhong, L., Bai, Z., Xia, H., Zhou, X.: Optimization and Application of Neural Network Modeling for Gray Problem. Computer project and Applications 37(8), 33–34 (2001)
14. Zhong, L., Bai, Z., Xia, H.: Whitening method of grey neural network modeling. Pattern Recognition and Artificial Intelligence 14(2), 145–149 (2001)
15. Shang, G., Zhong, L., Yan, J.: Establishment and Application of two Grey neural network model. Journal of Wuhan University of Technology 24(2), 78–81 (2002)
16. Zhong, L., Zhou, H.: Analysis of Oxidation heat treatment conditions' effect to electric properties of ceramics doping SrTiO3 based on grey neural network. Journal Wuhan University of Technology 26(1), 22–24 (2004)

On-Line Estimation of Biomass Concentration Based on ANN and Fuzzy C-Means Clustering*

Guohai Liu, Haixia Xu, Dawei Zhou, and Congli Mei

School of Electrical and Information Engineering, Jiangsu University, 212013, P.R. China
ghliu@ujs.edu.cn

Abstract. A multiple-model modeling method has been proposed for soft-sensing in a complex non-linear biochemical process for years. In this study, a multi-fuzzy-neural network model (MFNN), which is combined by multiple-model modeling method based on neural network and fuzzy c-means clustering algorithm (FCM), is presented to estimate the biomass concentration in fermentation process. Low dimensional sample data is achieved through principal component analysis (PCA). FCM is used for the analysis of the distribution of principal data and grouping them into overlapping clusters with different membership degrees. Then, a soft-sensing model is developed using multi-fuzzy-neural network to fit the different hierarchic property of the process. The biomass concentration is estimated by computing the sum of outputs of local models weighed by the corresponding degrees of membership. The model is applied to an erythromycin fermentation process, and case studies show that the approach has better performance compared to the conventional global model.

Keywords: soft-sensing, fuzzy c-mean clustering algorithm, neural network, fermentation.

1 Introduction

Over the last two decades, there has been a widespread development of biotechnological processes for industrial production and biotransformation. However, this development has been hampered by some important obstacles. A key obstacle is the lack of on-line sensors for key fermentation variables such as biomass and substrate concentrations [1]. The question of bioprocess control is how to monitor such concentrations in a reliable and low cost method [2]. To overcome this obstacle, a "software sensor" model has been proposed for fermentation systems [3]. Soft sensors are mathematical algorithms that unmeasured variables can be estimated from measured variables.

There are many different techniques for on-line estimation proposed in fed-batch reactors. Extensive reviews have been presented [4]. These techniques are based on either an empirical or mechanistic model of the fermentation process. However, in these methods, the functional form of the data-fitting model must be known in advance.

* This work was supported by Hi-Tech Research and Development Program (863) of China under Grant 2007AA04Z179.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 306–314, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Recently, the use of artificial neural networks (ANN) in modeling, identification and estimation of bioprocesses has been investigated by many researchers [5] [6] [7] [8], most of these works deal with the neural network as black-box models in the process. However, ANN-based model is based on the principle of empirical risk minimization (ERM), so the model based on ANN of nonlinear objects can't have good Robustness and Precision. In order to improve the model's Robustness and Precision, a new method that adding some models together was proposed by Bates and Granger in 1969. The method is to build some NN models respectively, then combine the outputs of the local models together to improve the model's Robustness.

With the inspiration of the multiple-model modeling method, a new method for soft-sensing modeling by clustering algorithm is presented. Bakker [9] [10] presented model clustering algorithm based on deterministic annealing, and the algorithm was applied to ensemble learning of neural network models. Giacinto [11] presented an algorithm for choosing individual neural networks based on clustering technique in ensemble learning. In this study, a new method for soft-sensing model is presented to solve the difficulties of the on-line measurement of biomass concentration in the erythromycin fermentation process. This proposed approach, MFNN model, is composed of three sub-networks: layer of feature extraction with PCA, fuzzy classifier with FCM and neural networks. Features of training data are extracted using PCA [12], the secondary variables are determined using PCA. After that, FCM is used for the analysis of the distribution of data and grouping them into overlapping clusters with different membership values and neural network is used for developing an optimal local model over each cluster. Finally, the estimation is implemented by computing the sum of the local models weighted by the corresponding membership degrees. The data collected from the erythromycin fermentation process are used to evaluate the effectiveness of the soft-sensing model, and the results show that the approach has better generalization ability and accuracy of estimation in fermentation process.

2 Material and Methods

For estimation of the concentration of biomass, different solutions were presented in the literature, such as the ANN model, self organizing map and some statistical analysis method. Fuzzy self-organizing layer, PCA and artificial neural network are combined to build a new model, named as the MFNN model, and then compare this technique with the conventional global model.

Low dimensional sample data are achieved by way of original data compression through principal component analysis (PCA), the outputs of PCA formed the input vectors to FCM, which is used to classified the input vector x into sets with different membership degrees. The overlapping clusters of the data space are more precise. The output data vectors of FCM are the input vector to every local neural network respectively.

2.1 Fuzzy c-Means Clustering Algorithms for Data Classification

Clustering, considered the most important unsupervised learning problem, which can discover intrinsic structures in a collection of unlabeled data and extract new knowledge from given data. There are many fuzzy clustering methods being introduced [13].

Fuzzy c-means clustering algorithm which proposed by Bezdek in 1981 is a variation of the popular k-means clustering algorithm, in which a degree of membership of cluster is incorporated for each data point [14]. The centroids of the clusters are computed based on the degrees of membership as well as data points.

For a set of unlabeled data

$$X = \{x_1, \cdots, x_N\},\ x_i = [x_{i1}, \cdots, x_{ik}]\ \ (i = 1,2,\ldots,N),$$

Where

N is the number of data points,
k is the number of input variables.

Its constrained fuzzy c-partition can be described as follows: Given the membership function of the ith ($i = 1,\ldots,N$) vector to the jth ($j = 1,\ldots,C$) cluster is denoted as μ_{ij} . The membership values are often constrained as

$$\forall i, \sum_{j=1}^{C} \mu_{ij} = 1; \forall i, j, \mu_{ij} \in [0,1]; \forall j, \sum_{i=1}^{N} \mu_{ij} > 0. \tag{1}$$

Then the quality of clustering can be measured by the objective function

$$J_m(U,V) = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^m \left\| x_i - v_j \right\|^2. \tag{2}$$

Where $V = \{v_1, v_2, \ldots, v_C\}$ is the vector of cluster centers and *m* is the weighting exponent. FCM can be summarized in the following steps:

Step 1: Determine the number of cluster, c and m-value (let m=2), given converging error, $\varepsilon > 0$ (such as $\varepsilon = 0.001$), appropriately choose the initial membership matrix $U^{(O)}$.

Step 2: Calculate the cluster centers V

$$v_i^{(l)} = \sum_{k=1}^{N} (\mu_{ik}^{(l)})^m x_k \,/ \sum_{i=1}^{N} (\mu_{ik}^{(l)})^m. \tag{3}$$

Step 3: Update the membership matrix U

$$\mu_{ik}^{(l+1)} = 1 / \sum_{j=1}^{C} (d_{ik} / d_{jk})^{2/(m-1)}, \forall i, \forall k \quad d_{ik} = \left\| x_i - v_k \right\|. \tag{4}$$

Step 4: Calculate the $J_m(U,V)$.

Step 5: Increment l ; until

$$\left\| J_m^{(l+1)} - J_m^{(l)} \right\| < \varepsilon. \tag{5}$$

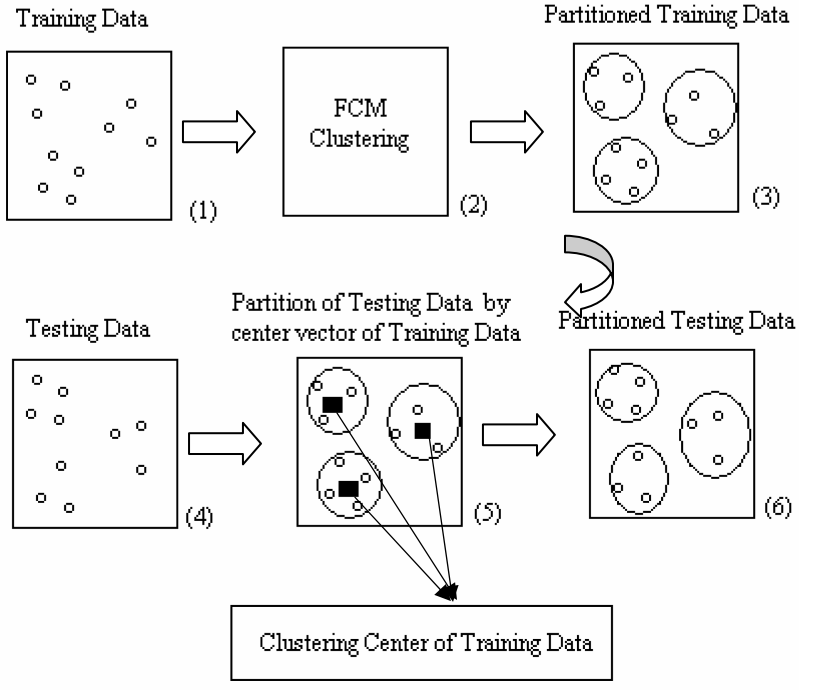


Fig. 1. Partition of training and testing data

Partitioned data are then used for the design of the individual neural networks. The partition of training and testing data is schematically visualized in Fig. 1. The partition of the training data is accomplished through phases (1)–(3) (Fig. 1). The way in which the testing data are processed on a basis of the previously constructed cluster centers is displayed in phase (5). Finally, the testing data set is partitioned as visualized by (6) in Fig. 1.

2.2 Multiple-Model Modeling Based on FCM (FMM)

Conventional data-based modeling methods focused on global approaches, such as neural networks, fuzzy set methods, and some other kinds of parametric models. However, when dealing with large data sets from industrial processes, traditional approaches become difficult on determination and optimization of efficient model structure. On the other hand, the ideal of multiple-model modeling is an approach that represents a nonlinear system with a set of simple local models in certain domains. FMM model is established by combining multiple-model modeling method based on ANN and fuzzy c-means clustering algorithm (FCM). At first, relevant sample data in each domain is clustered by FCM, then a local model is modeled based on relevant sample data in each domain, the output of the model is calculated by the local models at the final.

Considering of sample data of a MISO system can be expressed as:

$$D = \{Y, X_i | i = 1,2,\ldots,m\}$$

Where

$Y = \{y_i | i = 1,2,\ldots,n\}$ is the output matrix,

n is the number of sample data,

m is the number of input variables.

$\{X_i = [x_{1i}, x_{2i}, \ldots, x_{mi}]^T, i = 1,2,\ldots,m\}$, where X_i are the sample data of i input variables. If D is divided into c clusters, such as $\{D_1, D_2, \ldots, D_C\}$.We can build c local models $\{M_1, M_2, \ldots, M_C\}$ from the local sets $\{D_i | i = 1,2,\ldots,C\}$, the outputs of local models are $\mu_i, i = 1,2,\ldots,C$, μ_i calculated by FCM are the membership degrees of sample dates to cluster $1,2,\ldots,c$ respectively. The architecture of FMM is shown in Fig.2.

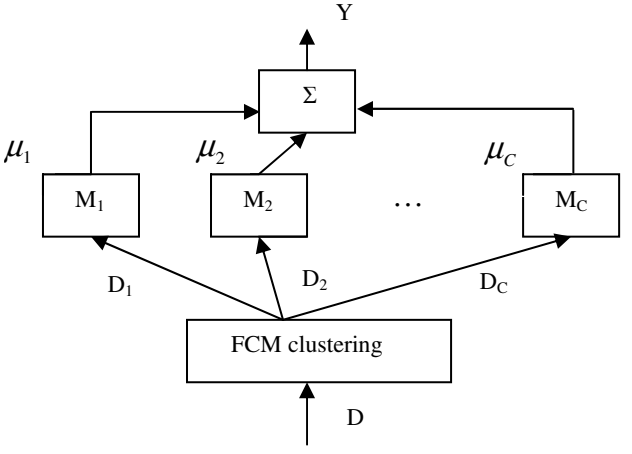


Fig. 2. Architecture of FMM

3 Model of Multi-fuzzy-neural Network (MFNN)

The architecture of the MFNN is based on a comprehensive and efficient framework of information classification produced with the aid of FCM and comes in the form of the fuzzy granulation formed throughout the partition of the space of input variables of each local NN. More specifically, we develop individual NNs on the basis of some clusters of data being constructed through FCM. The number of the clusters corresponds to the number of the models (NN structures). The functional details are outlined in Fig. 3. Noticeably, the flow of processing involves a fundamental decision as to the number of clusters (local models), its realization through the partition of the spaces, detailed design of the local models.

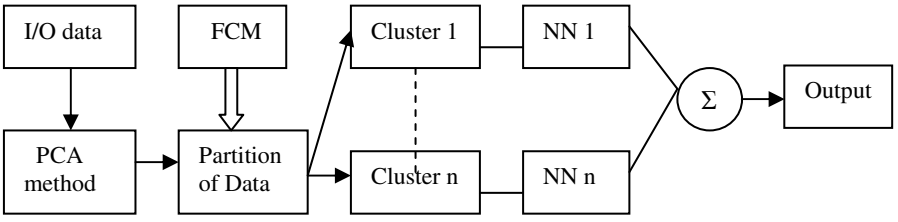


Fig. 3. Architecture of MFNN model

4 Soft-Sensing Model Based on MFNN

It is a typical modeling method to build a global soft-sensing model using ANN in the fermentation process [15]. However, the development of the model using a global approximation to estimate the biomass concentration would complicate the network and limit the accuracy of estimation due to the apparent differences about biomass concentration in different period of fermentation process.

The proposed soft-sensing model using the methods introduced in section 2 and section 3. At first, the data from the experiment of erythromycin fermentation process is preliminary analyzed with PCA, so the large multivariate data with highly correlated and noisy measurements can be compressed into a lower dimension space that contains most of the variance of the original matrix. The principal variables used as secondary variables to estimate the biomass concentration. Secondly, the number of clusters and some parameters in FCM are determined by priori knowledge, the sample data can be divided into four clusters for the analysis of the regulation of microbial growth in fermentation process. The variations of the microbial growth speed are relatively slow at the beginning, and then increase at the growth period, keep a relatively high speed at the intermediate period, drop a small range because of the inhibitors at the final. After the determination of the number of clusters, the soft-sensing model based on FMM can be modeled. The architecture of soft-sensing model is shown in Fig.4. Outputs of NNi

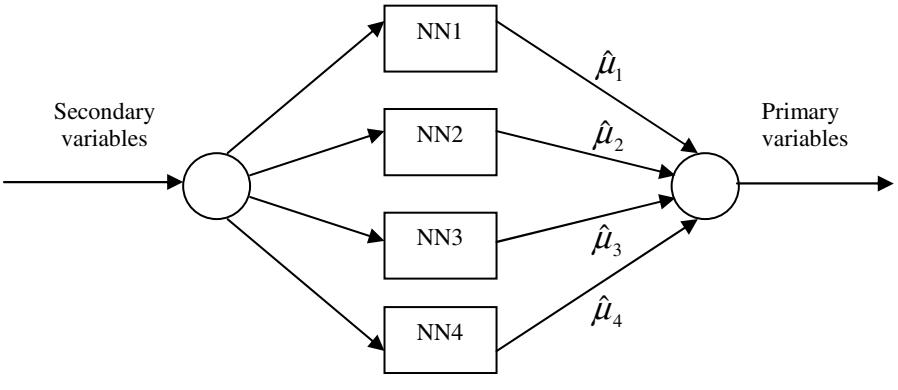


Fig. 4. Architecture of soft-sensing model

(i=1, 2, 3, 4) are $\mu_i (i = 1,2,3,4)$, and they are constrained by Eq. (1), so the following disposal is done.

$$\hat{\mu}_i = \mu_i / \sum_{j=1}^{4} \mu_j (j = 1,2,3,4). \tag{6}$$

The output of the model is calculated as follows.

$$\hat{X} = \sum_{i=1}^{4} \hat{\mu}_i X_i . \tag{7}$$

5 Case Study

All of the data used come from the experiment of erythromycin fermentation process. The whole erythromycin fermentation process lasts about seven days. In this work, about two hundreds points of raw data have been collected with physical sensors while the concentration of biomass is obtained by using off-line analyzers every six hours. The sample data can be classified into two parts: training data and test data. The batches 1-7 are used as training data sets, another three batches are used as test data sets. There are fifteen state variables in erythromycin fermentation. After the selection with PCA algorithm, the secondary variables such as temperature, PH, the relative pressure, agitator rotate speed and the biomass concentration at times t-1, dissolved oxygen as input variables are remained, the output variable of the model is biomass concentration. The structure of the each local neural network model is 6×10×1, and the hidden layer and output layer function of each neural network are sigmoid and linear function. The Levenberg-Marquardt back propagation (BP) algorithm is adopted to train the local models due to its faster convergence and memory efficiency. Three batches as test data are used to test the soft-sensing model, the test result is shown in Fg.5.

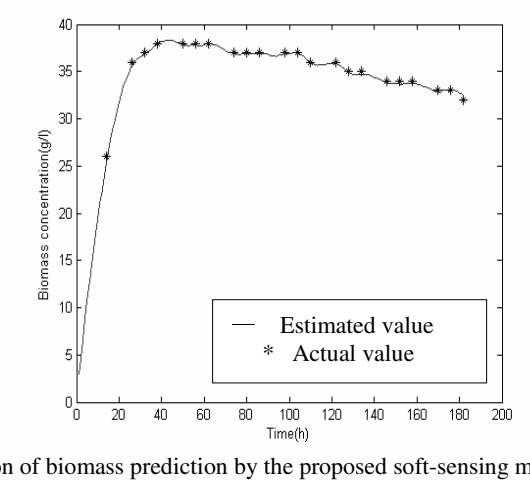


Fig. 5. Simulation of biomass prediction by the proposed soft-sensing model based on MFNN

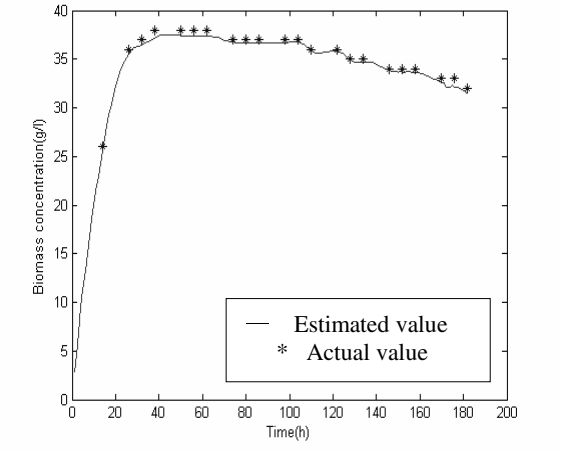


Fig. 6. Simulation of biomass prediction by the global model

Table 1. Testing errors of the two models

Test sets	The MFNN Model		Global Model	
	MRE	MES	MRE	MES
Sample 1	1.276%	0.1512	2.888%	0.4785
Sample 2	1.647%	0.2490	3.383%	0.5431
Sample 3	0.724%	0.1155	3.447%	0.5441

Mean square error (MSE) and maximum relative error (MRE) have been used as the accuracy of estimation to describe the effectiveness of the model prediction [16]. They are defined as:

$$\text{MRE}= \max\left(\frac{|y_i - \hat{y}_i|}{y_i}\right). \tag{8}$$

$$\text{MSE}= \sqrt{\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|^2}\ . \tag{9}$$

Where

N is the number of sample data pairs,

y_i is the measured value of biomass,

$\hat{y}_i$ is the corresponding estimated value predicted by the proposed model.

The outputs of the proposed model and a global soft-sensing model are compared to explain the performance of the soft-sensing model based on MFNN. The test result of global model is shown in Fig.6, and testing errors of the two models are listed in Tbl.1 (MSE, MRE).

The simulation results and the comparison of testing errors of the two models show that the approach proposed has better generalization ability. The accuracy of estimation of the soft-sensing model based on MFNN is improved in fermentation process.

6 Conclusions

A soft-sensing model based on MFNN is investigated to predict biomass concentration in erythromycin fermentation process. Relevant sample data in each domain are clustered by FCM, a local neural network model is modeled based on the relevant sample data in each domain. The proposed modeling method eliminates the need to train a large number of sample data at a time. The suitable topology of the soft-sensing model is evaluated via experimental data. From the results obtained in simulations, it can be concluded that the performance of the model established in this paper has indeed been improved to evaluate the on-line measurement of biomass concentration in the fermentation processes.

References

1. Leigh, J.R.: Modeling and Control of Fermentation Processes. Peter Peregrinus Ltd (1987)
2. Bastin, G., Dochain, D.: On-line Estimation and Adaptive Control of Bioreactors. Elsevier science, Amsterdam (1990)
3. Dochain, D.: State and Parameter Estimation in Chemical and Biochemical Process: A Tutorial. J. Proc. Cont. 13, 801 (2003)
4. Bastin, G., Van Impe, J.F.: Nonlinear and Adaptive Control in Bioreactors. Eur. Journal of Control, 37–53 (April 1991)
5. Craninx, M., Fievez, V., Vlaeminck, B., De Baets, B.: Artificial Neural Network Models of The Rumen Fermentation Pattern in Dairy Cattle. Computers and Electronics in Agriculture 60, 226–238 (2008)
6. Riverol, C., Cooney, J.: Estimation of The Ester Formation during Beer Fermentation Using Neural Network. Journal of Food Engineering 82, 585–588 (2007)
7. Tan, Z.P., Wang, S.T., Du, G.C.: Glutathione Fermentation Process Modeling Based On CCTSK Fuzzy Neural Network. Biotechnology 7, 73–79 (2008)
8. Patnaik, P.R.: Hybrid Filtering of Feed Stream Noise from Oscillating Yeast Cultures by Combined Kalman and Neural Network Configurations. Bioprocess and Biosystems Engineering 30(3), 181–188 (2007)
9. Bakker, B., Heskes, T.: Model Clustering by Deterministic Annealing. In: Proc. ESANN, pp. 87–92. D-Facto Public, Bruges (1998)
10. Bakker, B., Heskes, T.: Clustering Ensembles of Neural Network Models. Neural Networks 16, 261–269 (2003)
11. Giacinto, G., Heskes, T.: Design of Effective Neural Network Ensembles for Image Classification Purposes. Image and Vision Computing 19, 699–707 (2001)
12. Jin, H.D., Lee, Y.H., Lee, G., Han, C.H.: Robust Recursive Principal Component Analysis Modeling for Adaptive Monitoring. Industrial and Engineering Chemistry Research 45, 696–703 (2006)
13. Hoppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. Wiley Press, New York (1999)
14. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
15. Adilson, J., Rubens, M.: Soft Sensors Development for On-line Bioreactor State Estimation. Computers and Chemical Engineering 24(2), 1099–1103 (2000)
16. Willmott, C.J.: Some Comments on The Evaluation of Model Performance. Bulletin of American Meteorological society 63, 1309–1313 (1982)

PCA-Based Elman Neural Network Algorithm

Shifei Ding[1,2], Weikuan Jia[3], Chunyang Su[1], Xinzheng Xu[1], and Liwen Zhang[1]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221008 China
dingsf@cumt.edu.cn, dingshifei@sina.com
[2] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080 China
[3] College of Plant Protection, Shandong Agricultural University, Taian 271018

Abstract. When we manipulate high dimensional data with Elman neural network, many characteristic variables provide enough information, but too many network inputs go against designing of the hidden-layer of the network and take up plenty of storage space as well as computing time, and in the process interfere the convergence of the training network, even influence the the accuracy of recognition finally. PCA,which is short for principal component analysis is a multivariate analysis method which transforms many characteristic variables to few synthetic variables. Not only can it eliminate relationship among characteristic variables but new variables can also hold most information of the original ones as well. In this paper we make full use of the advantages of PCA and the properties of Elman neural network structures to establish PCA-Elman based on PCA. The new algorithm reduces dimensions of the high dimensional data by PCA, and carry on network training and simulation with low dimensional data that we get, which obviously simplifies the network structure, and in the process, improves the training speed and generalization capacity of the Elman neural network. We prove the effectiveness of the new algorithm by case analysis. The algorithm improves the efficiency in network problems solving and is worth further generalizing.

Keywords: Elman neural network, PCA, PCA-Elman algorithm.

1 Introduction

Artificial Neural Network (ANNhas the ability of large scale computing and has unique advantages when dealing with nonlinear high dimensional data[1]. Elman neural network is a kind of feedback neural network, that is to add a connecting layer to the hidden layer of feed-forward network as time delay operator so as to memorize and therefore, makes the system have the characteristic of time-varying and have stronger global stability[2]. When dealing with high dimensional data, useful information can be overwhelmed by large quantities of redundant data and relevant redundant information may take up much storage space and may be time consuming. What is more, too many characteristic inputs make the network structure complicated and therefore make training process time consuming and labour consuming, which harms the convergence of the network, and influence the accuracy of recognintion eventually. It is necessary to carry on pre-processing to the raw data of the problems of the Elman neural network

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 315–321, 2008.
© Springer-Verlag Berlin Heidelberg 2008

which need to be sovled. It is also necessary to analyze and extract useful characteristics of variables from large quantities of data and eliminate the influence of relevant and repeated factors, so as to reduce as many feature dimensions of the samples as possible without affecting solving the problems. The purpose is to simplify the structure of Elman neural network, improve training speed of the network, convergence and generalization capacity.

Combining dimension reduction of the high dimensional data with neural network organically is one of the hot spots in neural network algorithm improvement, which has made remarkable achievements. ES Gopi extracted characteristics of forged image by independent component analysis and identificate forged image by NN[3]. C Melchiorre carried on sensitivity analysis of landslide by combining clustering analysis with ANN, classified samples through distance measurement and in the process eliminated influence of relevant factors choosing significant classes so as to improve forecasting capacity of nueral network[4]. E Lewis and his staff analyze complex spectrum and time resolved measurements and determine the quality of the food[5] with the combination of PCA and neural network. But application of this kind is not common in Elman neural network, people like Yan Weiguo proposed a method of modeling based on PCA and Elman network to evaluate the quality of mechanical products [6] and apply it in predicting mechanical properties of cold-rolled ribbed steel wires and bars. People like Liu Boping combine PLS with Elman neural network and use near-infrared spectroscopy technique to carry on multi-component quantitative analysis to identificate its effectiveness and so on. Practical examples mentioned above have worked well but there is not a set of perfect algorithm theory to refer to in application.

From the feature effectiveness, PCA[8] is a method to represent the original multiple avariables with several comprehensive factors and make the comprehensive factors reflect as much of the information of the original variables as possible and the comprehensive factors do not relate to each other so as to reduce dimensions. Original raw high dimensional data, after being reduced by PCA act as inputs of Elman neural network to train network so as to get PCA-Elman, and to verify the feasibility of the new algorithm theory by analysis of instances.

2 PCA-Elman Combined Neural Network Algorithm

2.1 Basic Principle of PCA

First, raw data is normalized so as to eliminate the differences of index distribution, which can not only avoid repetition of the information but also overcome subjective factor in weight determination, and assure the utilization quality from data headstream. Suppose there are p variables in the sample, namely x_1, x_2, …, x_p they are integrated into p comprihensive variables by PCA analysis method, that is

$$\begin{cases} y_1 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1p}x_p \\ y_2 = c_{21}x_1 + c_{22}x_2 + \cdots + c_{2p}x_p \\ \cdots \\ y_p = c_{p1}x_1 + c_{p2}x_2 + \cdots + c_{pp}x_p \end{cases} \tag{1}$$

and they meet the following equation:

$$c_{k1}^2 + c_{k2}^2 + \cdots + c_{kp}^2 = 1 \quad (k=1, 2, \ldots, p) \tag{2}$$

In which y_i and y_j $(i \neq j\ ;\ i, j=1, 2, \ldots, p)$ are independent, y_1 is a linear combination satisfying the equations (1) which has the maximum variance of all the linear ones with characteristic variables, $x_1, x_2, \ldots, x_p$, y_2 goes next, and deduced by analogy, $y_p$has the smallest variance. These comprehensive factors of $y_1, y_2, \ldots, y_p$ are defined respectively as the first, second…pth principal components, and their variaces are degressive.

$$a = (\sum_{i=1}^{m} \lambda_i) / (\sum_{i=1}^{p} \lambda_i) \tag{3}$$

We can get p nonnegative eigenvalues of the correlation matrix of the sample with Jacobi's method and their relation is $\lambda_1 > \lambda_2 > \ldots > \lambda_p \geq 0$. Select m $(m<p)$ principal components, and we can see the proportion of the variance of the former m principal components taking up all of the variances can be defined as

$$a = (\sum_{i=1}^{m} \lambda_i) / (\sum_{i=1}^{p} \lambda_i) \tag{3}$$

Which approximates 1, therefore, the principal components of $y_1, y_2, \ldots, y_m$ basically embrace information of the original variables $x_1, x_2, \ldots, x_p$, and the number of variables has reduced from p to m, while these m variables contain the major original information and can reduce dimensions.

2.2 Fundamental Principle of Elman Nueral Network

Topology of Elman recurrent neural network is shown in chart1, in which the topology structure can be devided into four layers namely inpur layer, hidden layer, connecting layer and output layer. Connecting layer is used to memorize the output of the former moment from the hidden layer unit and can be regarded as a one-step time delay oprator. On the basis of the basic structure of BP neural network[9], the output of hidden layer is linked to the input of itself by the delay and storage of the output of hidden layer, which makes it sensitive to the data in historical states. The addition of interior feedback network increases the capability of processing dynamic information of the network itself. Storing the interior states makes it have the function of mapping dynamicity, and therefore makes the system have the ability to adapt to time-varying characteristics.

Suppose there are n inputs, m outputs and r neurons in hidden layer and connecting layer, the weight from input layer to hidden layer is w_1, while the weight between connecting layer and hidden layer is w_2 and the weight from hidden layer to output layer is w_3, $u(k\text{-}1)$ represents the inputs of the neural network, $x(k)$ represents the outputs of the hidden layer, $x_c(k)$ represents the outputs of the connecting layer, and $y(k)$ represents the outputs of neural network. Then

$$x(k) = f(w_2 x_c(k) + w_1(u(k-1))) \tag{4}$$

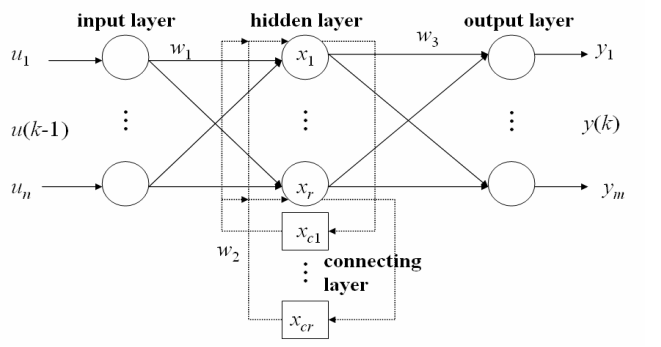


Fig. 1. Topology structure of Elman neural network

$$x_c(k) = x(k-1) \tag{5}$$

$$y(k) = g(w_3 x(k)) \tag{6}$$

In which f represents the transfer function of hidden layer. S type function is commonly used and can be defined as

$$f(x) = (1+e^{-x})^{-1} \tag{7}$$

g is the transfer function of the output layer and it is usually a linear function.
Elman network uses BP algorithm to modigy the weight values and the error of the network is:

$$E = \sum_{k=1}^{m} (t_k - y_k)^2 \tag{8}$$

In which t_k is the output vectors of the object.

2.3 Elman Neural Network Algorithm Based on PCA

When we use Elman neural network to deal with high dimensional data samples, too many network inputs cause a problem to the number of neurons in hidden layer and in the meantime can influence the network training, and influence identification precision in the process. If we reduce the dimensions of the raw data by PCA, so as to reduce the number of inputs, the designing of hidden layer will be more convenient and the network structure will be more simple. Let us establish PCA-Elman algorithm based on PCA（PCA-Elman algorithm）, the process of the algorithm is shown in chart2. High dimensional data samples can be reduced to low dimensional samples by PCA. The samples can be devided as training samples and simulation samples according to the need of the problem, and then training of the network and simulating and testing are carried on.

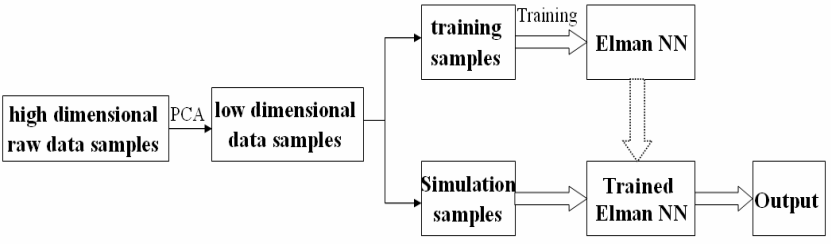


Fig. 2. Proceedure of PCA- Elman algorithm

Basic steps of PCA-Elman algorithm:

Step 1 Nomalize the raw data, and we get the mutual relation matrix *R* of the samples.
Step 2 compute the eingenvalue and eingenvector of *R* with Jacobi's method.
Step 3 compute the contribution rate and cumulative contribution rate, and select the number of principal components.
Step 4 compute the loads of principal components, get the scores of principal components finally and new low dimensional data, which can be regarded as novel samples ready to be dealt with.
Step 5 devide the new samples into two as training samples and simulation samples according to the need of the problem.
Step 6 design the structure of Elman neural network according to the number of principal components extracted and ouputs of practical problems.
Step 7 train network by using training samples, determine the number of neurons of hidden layer in Elman neural network and all of the bottom-up weights and threshold values.
Step 8 predict the simulation samples with trained network model.
The new algorithm improves the training speed and converge of the network by improving the structure of Elman neural network, and, in the process improves the effeciency in dealing with problems of the network.

3 Case Analysis

As data is known[10], we use meteorological factors to predict the occurrence degree of wheat midge. We select 50 samples from data of 1951-2000 as research objects and use x_1-x_{14} to represent 14 characteristic variables(meteorological factors) of the raw data ready to be dealt with, we also use Y to represent the occurrence degree of the wheat midge of the exact year. We use standardization to nomalize the raw data and still use X to represent the data we get. The system of prediction of wheat midge can be essentially regarded as an input-output system. The transformational relations include data fitting, fuzzy transformation and logic reasoning, which can all be expressed easily by neural network.

In order to explain the problems better, we use BP neural network, Elman neural network and PCA-Elman algorithm here to test the problem and compare these experimental results with each other. In the experiments we select 35 samples from

1951 to 1985 as training samples and select 15 samples from 1986 to 2000 as simulation samples. We list the results in table1.

Using MATLAB as a tool, we first establish a BP neural network with 14 neurons in the input layer and 1 neuron in the output layer and carry on simulation of the problem. Then we establish an Elman neural network with 14 neurons in the input layer and 1 neuron in the output layer to test it. In the process, we predict with PCA-Elman algorithm and analyze the sample data by PCA. Exact 6 principal components from the case, take scores of the principal components as the inputs of the network and establish a new Elman neural network.

Table 1. Comparison of BP algorithm, Elman algorithm and PCA-Elman algorithm

Network structure	Accuracy rate	Training epochs	Error square sum
BP algorithm	73%	441	12.1295
Elman algorithm	87%	412	4.9897
PCA-Elman algorithm	87%	294	3.2530

Table 1 shows that the predicting precision and convergence rate of Elman neural network are higher than those of BP network. Though information loss exists in the developed algorithm, which is used to reduce dimensions of raw data by PCA, the predicting precision of the new algorithm does not reduce, but the convergent steps reduce correspondingly, which shows that raw data, whose dimensions have been reduced, reduces the inputs of the network, that makes the designing of network more convenient and makes the structure of the network more simple, so as to improve the training speed of the network, showing that the improved algorithm is prior to the traditional algorithm.

4 Conclusion

In this paper, in order to overcome the shortcomings such as difficulties in designing hidden layers, time consuming and labor consuming when dealing with high dimensional data in Elman neural network, by combining Elman network with PCA, we propose the PCA-Elman neural network algorithm. Dimensions of raw data are reduced by PCA, which not only reduces data redundance but also eliminates influence of the relative factors. Elman neural network adds feedback function to BP neural network, is more capable in computing and has relatively better global network stabiligy. The new algorithm of PCA-Elman, which combines the advantages of PCA with those of Elman organically together can fit nonlinear prediction better. The results of analysis of insances show that although information loss exists in dealing with reducing dimensions of the raw data, the predicting precision does not reduce. The new algorithm improves the performance of network significantly, enhances the self-culture capabilities, speeds up the converge, saves run-time, and finally improves the operation efficiency of the network. The results of the new algorithm is satisfying, revealing its advantages of dealing with high dimensional data and is worth further generalizing.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant no.40574001, the 863 National High-Tech Program under Grant no. 2006AA01Z128, and the Opening Foundation of Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences under Grant no.IIP2006-2.

References

1. Mccllochw, S., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous activity. Bulletin of Mathematical Biophysics 10(5), 115–133 (1943)
2. Elman, J.L.: Finding Structure in Time. Cognitive Science 14(2), 179–211 (1990)
3. Gopi, E.S.: Digital Image Forgery Detection Using Artificial Neural Network and Independent Component Analysis. Applied Mathematics and Computation 194(2), 540–543 (2007)
4. Melchiorre, C., Matteucci, M., Azzoni, A., et al.: Artificial Neural Networks and Cluster Analysis in Landslide Susceptibility Zonation. Geomorphology 94(3), 379–400 (2008)
5. Lewis, E., Sheridan, C., Farrell, M.O., et al.: Principal Component Analysis and Artificial Neural Network Based Approach to Analysing Optical Fibre Sensors Signals. Sensors and Actuators A: Physical 136(1), 28–38 (2007)
6. Yan, W.G., Liu, G.X., Yang, J.: Quality Modeling of Mechanical Product Based on PCA and Elman neural network. Science Technology and Engineering 7(19), 5091–5095 (2007)
7. Liu, B.P., Qin, H.J., Luo, X.: Study on Quantitative Analysis of Multi-component Based on Near-Infrared Spectroscopy Technique in Elman Network. Spectroscopy and Spectral Analysis 27(12), 245–2459 (2007)
8. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis, 6th edn. Prentice Hall, Englewood Cliffs (2007)
9. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Representation by Back-Propagating Errors. Nature 3(6), 533–536 (1986)
10. Zhang, Y.M.: Application of Artificial Neural Network Used in Pridiction of Wheat Midge. Northwest Sci-Tech University of Agriculture and Forestry (2003)

Prediction Model of Water Resources in Mine Area Based on Phase Space Reconstruction and Chaos Neural Network

Keping Zhou[1], Ge Gao[1], Feng Gao[1], and Wenxiang Gao[2]

[1] School of Resources and Safety Engineering, Central South University, 410083 Changsha, China
[2] Yunnan Tin Industry Group Company, 661000 Gejiu, China
gf81412@126.com

Abstract. In the process of social economic development, water resource is increasingly scarce because of unreasonable exploitation of groundwater resources. It has seriously hampered the economic and social development speed in mine area, and even caused a series of negative effects of serious environmental and ecological problems. In this paper, chaos theory is used to study the water resource system in mine area. By analyzing the phenomena of chaotic characteristics in water resource system, regional mine water resources safety model was constructed based on the phase space reconstruction coupled with the neural network. Through the application of the model to forecast future water resources consumption in Gejiu mine area, the predicted results not only verified the validity of the model, but also found a new approach to study water resources in mine area.

Keywords: Water resources, phase space reconstruction, chaotic neural network, prediction.

1 Introduction

With the conflict between supply and demand of water resources getting worse, efficient prediction and rational allocation has become the most important means to realize the sustainable utilization of water resources. Gejiu tin mine, which located in southwest of Yunnan province of China, is a super-large metallic deposit dominated by tin and copper mineralization. In the eastern of the mine area, temporal and spatial variation distribution of water resources is imbalanced. Acutely changing of precipitation and runoff in annual and seasonal brings about many problems to development and utilization of water resources. It causes insufficient supply water in dry season, but difficult use in wet period and lead to flood disaster. The number of available water resources far below the water total resources and result in water wastage. Therefore, the development of Yunnan Tin Industry Group Company will be restricted due to imbalance distribution of water resources under the situation of production distribution adjusting and rapid economic development.

At present, semi-empirical, conceptualize or random theory mainly be used on the study of macro and meso-scale water resources system. However, the hydrological process in objective world is the complex phenomena intertwined various components

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 322–330, 2008.
© Springer-Verlag Berlin Heidelberg 2008

of certainty and uncertainty. In recent years, research of chaos theory has made significant progress. People have come to realize that chaos is a widespread natural phenomenon and it has been gradually penetrate into many branches of science. Of course, research in the field of water resources system is also one of new directions.

By analyzing the chaotic phenomena in water resources system, phase space reconstruction coupled with neural network of regional mine water resources safety model was constructed. Application of the model shows that we can predict the dynamic development process of water resources system effectively.

2 Chaos Theory

By far, the study of the chaos by nonlinear dynamics is most systemic and stringent. In nonlinear dynamics, a number of definitions for theories determine and scale for actual measurement were put forward only on the consideration of perspective of mathematics and physics. However, it has lays a foundation to the establishment and development of chaos. Li-Yorke theorem is a greater impact and more accepted mathematical definition of chaos.

Li Tianyan and Yorke • JA first proposed the word "chaos" in 1975. They presented the chaos of a mathematical definition in the article "cycle of 3 means that chaos", which has known as Li-Yorke definition in nowadays.

Li-Yorke theorem: let $f(x)$ is continuous self-mapping on interval [a, b], if $f(x)$ has 3 periodic points, $f(x)$ has n periodic points to any positive integer n.

Chaos definition: let continuous self-mapping f: $I \to I$ contained in R, I is a subinterval of R, if set I is included in uncountable set S and satisfy:

(1) S contains on periodic points.
(2) If any X_1, $X_2 \in S$ ($X_1 \neq X_2$), we have

$$\limsup_{t\to\infty} \left| f^t(X_1) - f^t(X_2) \right| > 0 \tag{1}$$

$$\liminf_{t\to\infty} \left| f^t(X_1) - f^t(X_2) \right| = 0 \tag{2}$$

where, $f^t(X)=f(f\cdots f(X))$ represent t multiple function relationship.

If any $X_1 \in S$ and arbitrary periodic point of f $P \in I$, then we have

$$\limsup_{t\to\infty} \left| f^t(X_1) - f^t(P) \right| > 0 \tag{3}$$

We define that f in S is chaos.

3 Phase Space Reconstruction and Chaos Neural Network Model

3.1 Phase Space Reconstruction of Chaotic Time Series

The basic idea of phase space reconstruction is that the information of system state in time series is emerged by reconstruction technology. The study of chaotic time series

signal processing methods focus on the perspective of dynamic system. It is different to traditional approach that focuses on various transformations and processing of time series. In order to obtain the information of system from time series, Pachard et al. presented the phases pace reconstruction technology at first, then it has further developed and proved by Taken by using mathematics, and the theorem has known as the phase space reconstruction (Takens theorem).

The n dimensional dynamic system (a group has n one order differential equations) is given as

$$\frac{dx_i}{dt} = f_i(x_1, \cdots, x_n) \qquad (i = 1,2, \cdots, n) \tag{4}$$

which can be changed into an n order alignment differential equation by elimination method

$$x^{(n)} = f(x, x^{(1)}, \cdots, x^{(n-1)})$$

The orbit of system after changing is expressed as

$$\vec{x}(t) = [x(t), x'(t), \cdots, x^{(n-1)}(t)]$$

It describes the same dynamics problem and evolves in phase space formed by coordinate $x(t)$ adding $(n-1)$ order derivative $x'(t),\ldots,x^{(n-1)}(t)$. Instead of this continuous variable $x(t)$ and its derivative, we can consider discontinuous time series and its $(n\text{-}1)$ delay displacement:

$$\vec{x}(t) = \{x(t), x(\tau), \cdots, x[t + (n-1)]\}$$

Choose delay τ as long scale study object of time series, if dynamic system exist single state variables and it orbit evolution evolves in the phase space which formed by time displacement coordinates, the attractor of dynamic system can be reconstructed in a new phase space with delay coordinates which no change in its topological feature.

In practical application, for given time series: $x(t_0)$, $x(t_1)$, …, $x(t_i)$,…, $x(t_n)$, it is usually extended to three-dimensional or higher, and this is the delay in state-space coordinates reconstruction. Continuation of the above time series will become to space phase distribution in n-dimensional space as

$$\begin{bmatrix} x(t_0) & x(t_1) & \cdots & x(t_i) & \cdots & x(t_n-(m-1)\tau) \\ x(t_0+\tau) & x(t_1+\tau) & \cdots & x(t_i+\tau) & \cdots & x(t_n-(m-2)\tau) \\ x(t_0+2\tau) & x(t_1+2\tau) & \cdots & x(t_i+2\tau) & \cdots & x(t_n-(m-3)\tau) \\ \vdots & \vdots & & \vdots & & \vdots \\ x(t_0-(m-1)\tau & x(t_1-(m-1)\tau & \cdots & x(t_i-(m-1)\tau & \cdots & x(t_n) \\ X(t_0) & X(t_1) & \cdots & X(t_i) & \cdots & X(t_0') \end{bmatrix} \tag{5}$$

where τ=k$\triangle$t (k=1, 2, …, n) is delay time, and each column in the above formula form a phase point of m-dimensional phase space. Any phase point $x(t_i)$ has m-dimensional components $[x(t_i), x(t_i+\tau), \ldots, x(t_i-(m-1)\tau)]$. The phase point of

t_i- (m-1)τ forms a contour in m-dimensional phase space, and the lines between points descript the evolution orbit of system.

3.2 Selection of Phase Space Reconstruction Parameters

In order to construct phase space from time series, except determining m, we also need to give proper sampling interval τ. Theoretically speaking, the selection of τ is nearly arbitrary. However, in actual systems, the value of τ is determined by trying again and again. If τ is too small, the adjacent value x_i and x_i+x are close to each other, so they can not independent, and the orbit of phase space is a straight line. Contrary, if τ is too large, it makes the data point concentrate in a small area and we can not obtain attractor's local structure from reconstruction of the phase diagram. In the reconstruction phase space time delay τ and dimension are both important, and they are difficult to select.

C-C method is presented by R.Eykholt and J.D.Salas in 1999. This method use correlation integral to estimate the delay of time series and delay time window. Let Y_i is a point in phase space, then the correlation integral embedded in time series can be definition as follows:

$$C(m,N,r,t)=\frac{2}{M(M-1)}\sum_{1\le i\le j\le M}H(r-\left\|Y_i-Y_j\right\|),r>0 \tag{6}$$

where m is embedded dimension; N is series size; M=N-(m-1)τ; H is Heaviside function.

To ordinary time series, which divided into t disjoint subsequences, then $S(m,N,r,t)$ of each subsequence is defined as:

$$S(m,N,r,t)=\frac{1}{t}\sum_{s=1}^{n}C(m,\frac{N}{t},r,t)-C_s^m(1,\frac{N}{t},r,t) \tag{7}$$

Let $N\to\infty$, and choose the two corresponding max and min radius r, defined the dispersion as follows:

$$deltS(m,t)=\max S(m,r_j,t)-\max S(m,r_j,t),m=2,3,\cdots$$

According to the statistical conclusions of BDS, take m=2,3,4,5, r_j=$j\delta$/2, j=1,2,3,4, where δ is square deviation, then we have

$$S(t)=\frac{1}{16}\sum_{m=2}^{5}\sum_{j=1}^{4}S(m,r_j,t) \tag{8}$$

$$deltS(t)=\frac{1}{4}\sum_{m=2}^{5}deltS(m,t) \tag{9}$$

$$Scor(t)=S(t)+\left|deltS(t)\right| \tag{10}$$

One can make map based on the above result, taking time delay corresponding with t of $\overline{delt S}(t)$'s first minimum, or corresponding with t of $S(t)$'s first zero point, or corresponding with t of $Scor(t)$'s minimum.

4 Application of Model

Forecasting future water demand of regional mine is the base of mine development. Carrying on the accurately forecast to future short-term water load is to ensure that the mine area's water supply safety and meet the needs of economy operation

Table 1. Water consumption of Gejiu mine area in the past 30 years

Year	Water consumption ($\times 10^4$ m^3)					
	Feb.	Apr.	May	Aug.	Oct.	Dec.
1977	58.50	60.90	57.43	142.76	194.60	137.38
1978	305.52	65.33	77.60	138.54	138.55	115.45
1979	84.17	51.12	200.46	88.28	132.36	144.98
1980	277.73	154.24	53.72	74.03	338.15	121.78
1981	54.09	57.67	58.84	105.00	95.48	154.15
1982	116.32	77.97	94.28	85.71	122.79	116.13
1983	104.24	125.93	120.27	148.76	174.44	118.04
1984	147.62	140.04	140.61	138.90	155.79	127.43
1985	118.83	167.09	189.72	145.88	167.04	163.39
1986	139.11	155.98	165.18	129.30	129.43	138.58
1987	208.37	220.43	185.43	174.18	157.51	110.29
1988	129.60	139.95	112.01	132.37	160.18	235.51
1989	172.76	118.44	139.11	171.12	261.41	104.61
1990	149.54	168.12	129.89	160.64	187.87	143.29
1991	182.73	135.12	138.04	141.81	155.99	98.40
1992	89.52	141.42	141.44	137.53	105.65	73.62
1993	86.06	97.63	85.73	100.71	82.17	101.58
1994	97.29	98.05	78.32	88.02	83.96	72.25
1995	84.95	95.38	94.18	77.02	93.99	86.17
1996	90.86	86.09	89.77	85.70	74.69	86.84
1997	86.69	58.43	71.89	42.47	54.22	75.47
1998	63.44	55.51	70.68	39.78	68.85	81.89
1999	64.64	67.96	62.77	31.48	54.27	58.26
2000	99.16	136.41	96.96	95.46	88.74	153.41
2001	98.87	256.06	87.54	67.2	74.47	89.27
2002	85.33	100.65	226.08	68.69	82.89	158.98
2003	82.45	89.66	138.87	62.84	75.2	93.73
2004	182.95	82.77	86.26	150.78	95.39	476.33
2005	192.44	100.95	190.98	70.02	199.82	111.18
2006	113.68	123.1	109.84	89.63	91.05	147.65

scheduling. Since there are many factors affecting water demand, it is hardly use function model to describe. Therefore, in order to achieve science and accurate forecast of water consumption, a simple and practical water demand forecast model must be established. Table 1 is the water consumption of Gejiu mine area in the past 30 years.

According to actual data in Table 1 after model established, we ensure that the network structure contains one input layer, two hidden layer, and an output layer, which showing as Fig.1. Extracting a sample of 25 from table1 as training set, then training the grid, and choosing 7 samples as validation set, taking the validation error accuracy is 0.01. Fig.2 is the error change process while grid training and validation.

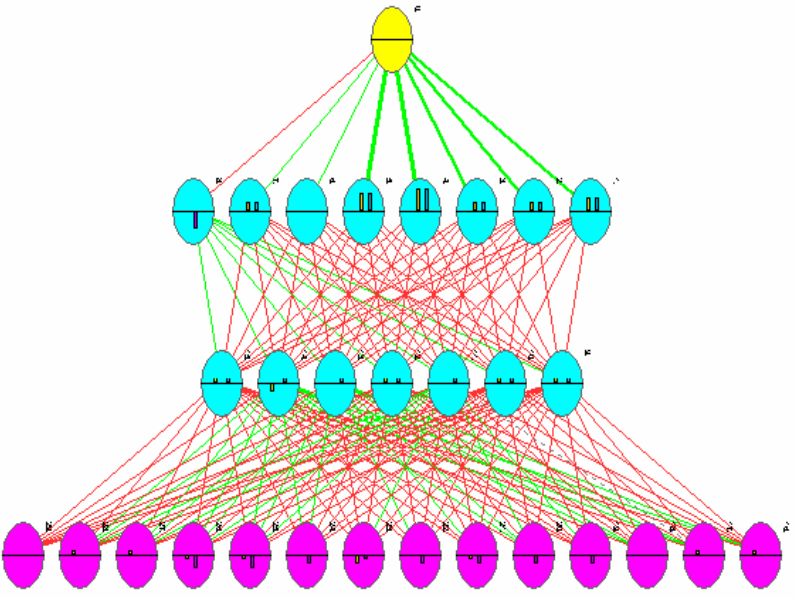


Fig. 1. Neural network structural diagram

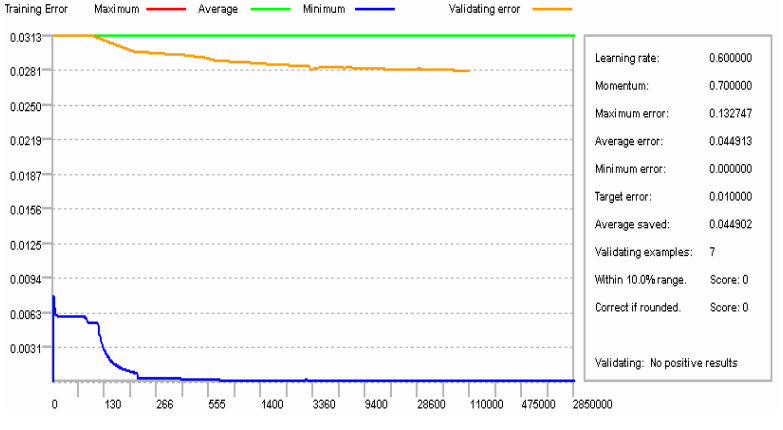


Fig. 2. Network study validation error convergence process curves

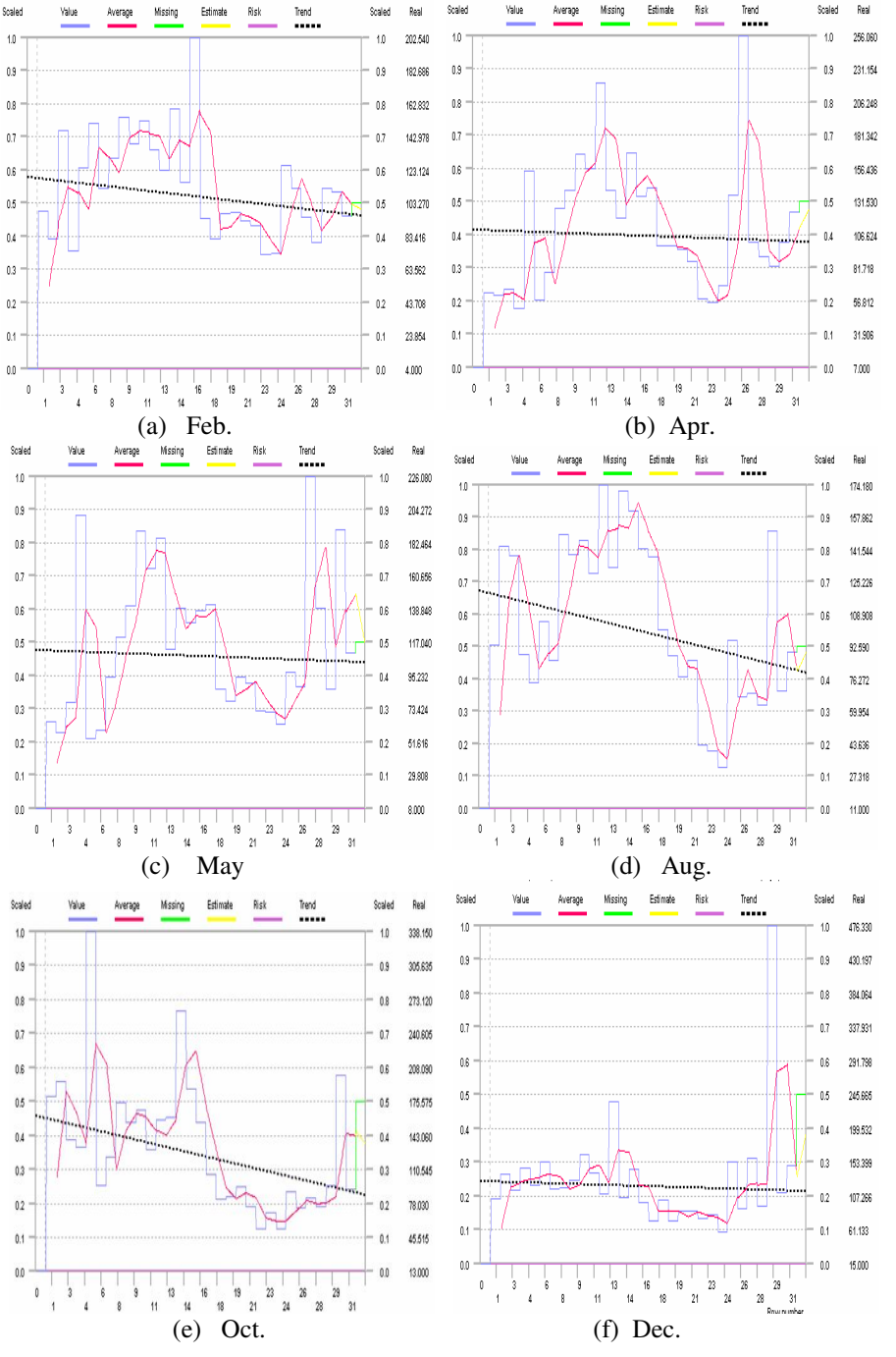


Fig. 3. The water resources development change curve in different period of mine area

After the success training, adding the last (m-1)-dimensions phase space to the network input, then can achieve the forecast of $x(t_n+1)$, while the phase space reconstruction coupled with the neural network forecast model has established. Based on this model, prediction of mine area water demand will be achieved. Fig.3 shows the water resources development and change process in different periods of mine area.

Fig.3 shows the water resources change curve in February, April, May, August, October and December of the mine area. From the pictures, we can see that water resources of Gejiu mine area is reducing in the total tendency. Water resources remain relatively stable in April or May and December, and it has big fluctuations in other months. There are two main aspects which cause this phenomenon. One is natural conditions, as the monthly rainfall, water storage capacity will be different in a year, and the impact of geological conditions in the region, which making some months' changes in the trend of relatively slow and others' changes magnitude larger. The other is artificial conditions, such as the excessive exploitation of underground water and irrational development. Furthermore, the rapid development of industrial and agricultural also causes the use of industrial water increase significantly. All of these will lead to steady decline in water reserves. Therefore, we must further strengthen the optimal allocation of water resources, improving the utilization of water resources in mine area.

Through establishing water demand forecast model for Gejiu mine area, we can effectively predict the changes of mine's total water resources and provide the basis for the relevant departments. Aim at current actual situation and the forecast results at an early stage and formulating a reasonable water resources exploitation policy. Truly achieved the best of water resources development, management, protection, and making it bring into play the best economic benefit in promoting the ecological environment and socio-economic coordinated and healthy development.

5 Conclusions

(1) Because of multitudinous and stochastic characteristics of influence factors on water demand in mine area, it is hard to use function model to forecast water consumption. This article analyzed the chaos characteristics of water resources system and introduced chaos theory in the security area of water resources. By Combining with chaos theory and neural network, a new ideal for safety forecast and optimal allocation of water resources is open up.

(2) Through model debugging, error and sensitivity parameters analyzing, the phase space reconstruction coupled with the neural network of regional mine water resources safety model was constructed according to chaos theory of space reconstruction principle. Compared the simulation value with historical statistical data, the validity of the model was validated. Hence, characteristics of water resources bearing capacity in eastern Gejiu mine area can be reflected, and the dynamic development process can be effectively forecasted.

References

1. Liu, S.Y., Zhu, S.J., Yu, X.: Determinating the embedding dimension in phase space reconstruction. Journal of Harbin Engineering University 4, 374–381 (2008)
2. Xiao, F.H., Yan, G.R., Han, Y.: Information theory approach to determine embedding parameters for phase space reconstruction of chaotic time series. Acta Physica Sinica 2, 550–556 (2005)
3. Yilmaz, L.: Chaos in the water resources system. Applied Mathematics and Computation 2, 761–773 (2006)
4. Sua, Z.-Y., Wub, T., Yangc, P.-H., Wang, Y.-T.: Dynamic analysis of heartbeat rate signals of epileptics using multidimensional phase space reconstruction approach. Physica A: Statistical Mechanics and its Applications 10, 2293–2305 (2008)
5. Deng, J., Yue, Z.Q., Tham, L.G., Zhu, H.H.: Pillar design by combining finite element methods, neural networks and reliability: a case study of the Feng Huangshan copper mine, China. International Journal of Rock Mechanics and Mining Sciences 4, 585–599 (2003)
6. Lo, E.Y., Mei, C.C.: Slow evolution of nonlinear deep water waves in two horizontal directions: A numerical study. Wave Motion 3, 245–259 (1987)
7. Urban, L.V., Templer, O.W.: Water resource management on the Texas High Plains: controlled chaos or a model of efficiency? Management of Irrigation and Drainage Systems: Integrated Perspectives, 116–123 (1993)
8. Yang, X.H., Li, J.Q.: Source: Application of chaos real-encoded genetic algorithm in water quality model parameter optimization. Water Resources and Power 5, 1–4 (2006)
9. Napiorkowski, J.J., Terlikowski, T.: Application of deterministic chaos and neural networks in water reservoir management. In: AIP Conference Proceedings, vol. 573, pp. 349–359 (2001)
10. Liu, Y.J., Shen, J.S.: Prediction model of urban water consumption based on chaos neural network. Water Resources and Power 1, 15–27 (2005)
11. Cao, L.H., Hao, S.L., Chen, N.X.: Study on resource quantity of surface water based on phase space reconstruction and neural network. Journal of Coal Science and Engineering 1, 39–42 (2006)
12. Sivakumar, B., Jayawardena, A.W., Fernando, T.M.K.G.: River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches. Journal of Hydrology 30, 225–245 (2002)

Predictive Control Strategy of Hydraulic Turbine Turning System Based on BGNN Neural Network

Yijian Liu[1,2] and Yanjun Fang[1]

[1] Department of Automation, Wuhan University, 430072 Hubei, China
Liuyijian_2002@163.com
[2] School of Electrical & Automation Engineering, Nanjing Normal University, 210042 Jiangsu, China
yjfang@whu.edu.cn

Abstract. A model predictive control (MPC) strategy based on a novel Bayesian-Gaussian neural network (BGNN) model was proposed for the controller design of hydraulic turbine in this paper. The BGNN was used to learn the nonlinear dynamic model of controlled hydraulic turbine on-line as the predictive model for the design of MPC controller. Experiments show that the proposed nonlinear MPC strategy based on BGNN performs much better than the conventional PID controller.

Keywords: Hydraulic Turbine, BGNN, Predictive Control, neural network.

1 Introduction

The hydraulic turbine system (HGS) is an essence complex and nonlinear system with time-varying [1]. The control strategy plays an important role in the steady running of the HGS. The traditional control strategy for HGS is PID controller which parameters are attained under a certain operating mode [2]. Owning to the operating situation changed often, the PID controller could not maintain good dynamic performance in the whole operating range of HGS. So it is necessary to design good controllers according to the change of operating situation of the HGS.

Model predictive control (MPC) technique for nonlinear system has been recognized as a powerful methodology for controlling a wide class of nonlinear dynamic systems [3][4][5]. The innovative feature of the method is that the controller design procedure is cast as an optimization problem. In the design of MPC controller the predictive model of controlled object is the key problem and the neural networks are often used for the nonlinear dynamic process model because the neural networks have been proved that they can approximate any nonlinear systems [6][7].

The Bayesian-Gaussian neural network (BGNN) is a posteriori probability model based on Bayesian theory and Gaussian hypothesis first proposed in the reference [8]. The BGNNs have been successful applied for the model identifications of nitrogen oxides (NOx) emissions of power plant boiler [9]. In this paper, the BGNN is used to learn the nonlinear model of hydraulic turbine first. Then the MPC controller based on the BGNN predictive model is designed.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 331–341, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The paper is organized as follows. A brief description of HGS follows this introduction. In section 3, the whole control strategy is described. The section 4 contains the BGNN architecture considered here and the generation of the training data set. Section 5 describes the design of the model predictive controller based on BGNN predictive model. Simulation 6 examines the performance of the model predictive controller based on BGNN and compares it with a traditional PID controller in HGS. Section 7 draws some conclusions and presents suggestions for future improvement.

2 Description of the HGS

The hydraulic turbine system (HGS) includes the servomechanism, penstock system, water turbine and generator illustrated in Fig. 1.

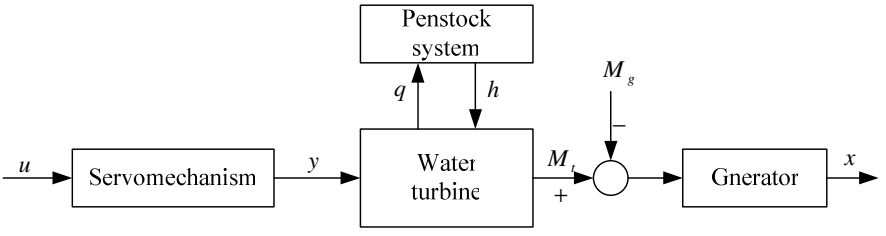


Fig. 1. Illustration structure of hydraulic turbine system

$$y = \frac{1}{1+T_y s}u \tag{1}$$

The servomechanism can be approximately expressed as a first order equation with its translating function model described in formula (1). T_y is inertia time constant of servomechanism and y denotes the output of servomechanism.

In dynamic process, the characteristics of water turbine running vary with the change of operating condition. The movable Francis turbine is nonlinear in essence. The nonlinear characteristics of movable Francis turbine can be depicted as the equation (2) as follows:

$$\begin{cases} m_t = f(y,x,h) \\ q = q(y,x,h) \end{cases} \tag{2}$$

where q is flow and m_t denotes the movable turbine moment. h is water head and x is the speed of rotation (generally x is expressed as frequency). The function relations of f and q are nonlinear and difficult to obtain through mathematical analysis method. Neural network as a black modeling technique can be used to the nonlinear model identification. So BGNN in this paper is suitable for the above nonlinear characteristics modeling of the HGU.

The dynamic characteristic of penstock system is complex and nonlinear between the water head h and flow q. Due to the difficulty of establishing the precise nonlinear

math model, BGNN is also used to learn the nonlinear relation listed as the following equation (3).

$$h = h(q) \tag{3}$$

Dynamic equation of generator taking account of load characteristics is often simplified as the follows:

$$x = \frac{1}{T_a s + e_n}(m_t - m_g) \tag{4}$$

where T_a is inertia time constant of generator and e_n denotes the adjusting coefficient of load. Those parameters vary with water turbine wok situation and need to be identified.

3 Overall Control Strategy

The control structure of the HGU is illustrated in Fig.2. Where x_{sp} is the reference input signal. $u(t)$ is the output of the MPC and the $\tilde{x}(t)$ is the BGNN model output at the time *t*.

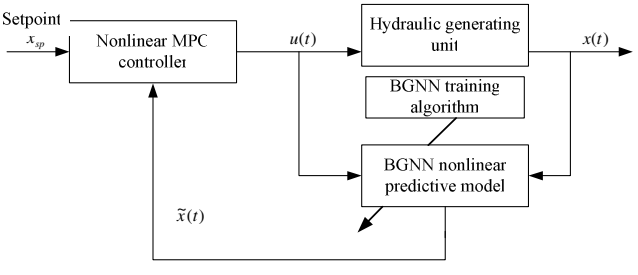


Fig. 2. The nonlinear MPC structure of the HGS

The control strategy of the HGS control system is composed of the controlled object HGS and nonlinear MPC controller. The BGNN is used to identify the input and output relation of the HGU which will be acted as the predictive model in the design of the MPC controller which will be described in the next sections.

4 BGNN Neural Network

The Bayesian-Gaussian neural network (BGNN) is first introduced as a posteriori probability model which structure adopts some concepts of neural networks. The advantages of the BGNN over the traditional neural networks such as BP neural work and RBF neural networks are easy definition of network topology and less training time of weights of network [8].

4.1 The Topology of the BGNN

The BGNN topology is shown in Fig. 3 and the structure of neural node is revealed in Fig.4. The node of the BGNN is an ordinary neural node just as in the other networks. The activation function of the node varies in different layers or even in the same layer. The every layer will be described in the following.

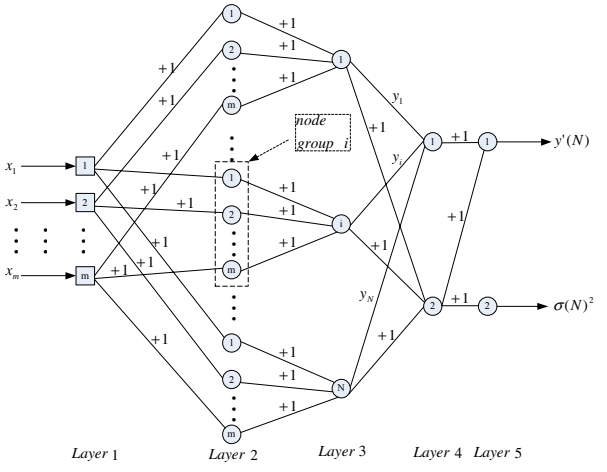


Fig. 3. The topology and connection weights of the BGNN

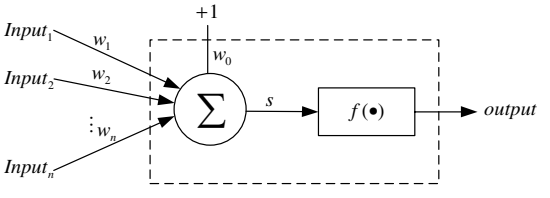


Fig. 4. The universal neural node

Layer 1: Simply store the *m* input values.

Layer 2: *N* groups of nodes and the *i*th group corresponds to the *i*th sample in the training data set. Each group has m nodes. For the *j*th node in the *i*th group

$$s_{ij}^{[2]} = x_j, f^{[2]}(s_{ij}^{[2]}) = \frac{(s_{ij}^{[2]} - x_{ij})^2}{d_{jj}^2} \tag{5}$$

The Superscript [2] stands for the second layer. Similarly, [3] and [5] below stand for layers 3and 5, respectively.

Layer 3: *N* nodes. The *i*th node corresponds to the *i*th sample in the training data set. For the *i*th node

$$s_i^{[3]} = \sum_{j=1}^{m} \frac{(x_j - x_{ij})^2}{d_{jj}^2}, f^{[3]}(s_i^{[3]}) = \sigma_0^{-2} e^{-s_i^{[3]}} = \sigma_i^{-2} \tag{6}$$

Layer 4: 2 nodes. For the 1st and the 2nd node

$$s_1^{[4]} = \sum_{i=1}^{N} y_i \sigma_i^{-2}, f_1^{[4]}(s_1^{[4]}) = s_1^{[4]} \tag{7}$$

$$s_2^{[4]} = \sum_{i=1}^{N} \sigma_i^{-2}, f_2^{[4]}(s_2^{[4]}) = s_2^{[4]} \tag{8}$$

Layer 5: 2 nodes. For the 2nd and the 1st node,

$$s_2^{[5]} = \sum_{i=1}^{N} \sigma_i^{-2}, f_2^{[5]}(s_2^{[5]}) = \frac{1}{s_2^{[5]}} = \sigma(N)^2 \tag{9}$$

$$s_1^{[5]} = \sum_{i=1}^{N} y_i \sigma_i^{-2}, f_1^{[5]}(s_1^{[5]}) = \frac{s_1^{[5]}}{s_2^{[5]}} = y'(N) \tag{10}$$

From the description of the above layers, the input and output relations can be represented as the formulas listed in equation (11) to equation (14).

$$p(Y \mid Y_1, Y_2, ..., Y_N) = \frac{c}{\sqrt{2\pi}\sigma(N)} e^{\frac{-1(Y - y'(N))^2}{\sigma(N)^2}} \tag{11}$$

where c is a normalizing constant.

$$y'(N) = \sigma(N)^2 \sum_{i=1}^{N} \sigma_i^{-2} y_i \tag{12}$$

$$\sigma(N)^{-2} = \sum_{i=1}^{N} \sigma_i^{-2} \tag{13}$$

Assume that

$$\sigma_i^{\ 2} = \sigma_0^2 e^{(X - X_i)^T D (X - X_i)} \tag{14}$$

where D is input threshold matrix and d_{11}, d_{12},…, d_{mm} are named the input factors, which will be evaluated through the network training.

$$D = \begin{bmatrix} d_{11}^{-2} & & \\ & d_{jj}^{-2} & \\ & & d_{mm}^{-2} \end{bmatrix} \tag{15}$$

The criterion of minimize is similar the Prediction Error method:

$$V_N(D) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - y_i^{'})^2 \tag{16}$$

where N is the net order, while y_i and $y_i^{'}$ are denoted as the desired output and the network output for sample i respectively.

The off-line training of the BGNN is to selected proper matrix D in order to minimize the criterion function $V_N(D)$. In our work, the downhill simple optimization is employed to obtain the parameters in D.

4.2 Self-tuning of BGNN Model and On-Line Identification Procedure of the HGS Nonlinear Model

Differing form the traditional neural works such as the BP neural network and RBF neural work, the BGNN owns the advantage of self-tuning ability and adapts to the on-line model identification application. Therefore the BGNN is used to the nonlinear on-line identification of HGS as the predictive model of the MPC controller.

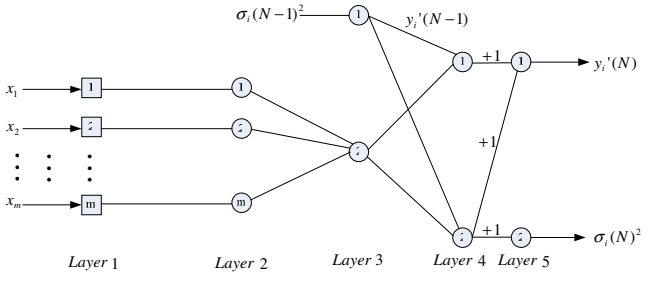


Fig. 5. Schematic of the recursive BGNN

In the self-tuning process, it is also necessary that a training sample be exclude from the training sample set and the effects of this sample be eliminated which may also be calculated through a recursive model listed in equations (17) to (19). Suppose the effects of the k*th* training sample has been eliminated from the sample set, then

$$\sigma_k^{\ 2} = \sigma_0^2 e^{(X-X_k)^T D(X-X_k)} \tag{17}$$

$$\sigma_k(N-1)^{-2} = \sigma(N)^{-2} - \sigma_k^{\ -2} \tag{18}$$

$$y'_k(N-1) = \sigma_k(N-1)^2(\sigma(N)^{-2}y'(N) - \sigma_k^{\ -2}y_k) \tag{19}$$

Suppose the BGNN order has been trained off-line with net order N. At each time instant, a new training sample is added to the training data set and in the meantime one sample is to be deleted from these N+1 training samples. Therefore it is necessary to determine which sample should be deleted form the N+1 training samples. The

measure of mean square prediction error (MSPE) is defined in order to evaluate every training sample. The MSPE of the i*th* training sample is shown in formula (20)

$$MSPE_i = E(y_i - Y_i(N))^2 = (y_i - Y_i(N))^2 + \sigma_i(N)^2 \tag{20}$$

If a training sample contains little innovation, it is highly possible that the output at this point can be predicted with high accuracy by other samples. Therefore the sample which has the least MSPE should be eliminated from the set of *N*+ 1 sample.

The BGNN on-line application can be divided two procedures: off-line learning procedure and self-tuning process. The program flow chat is shown in Fig.6.

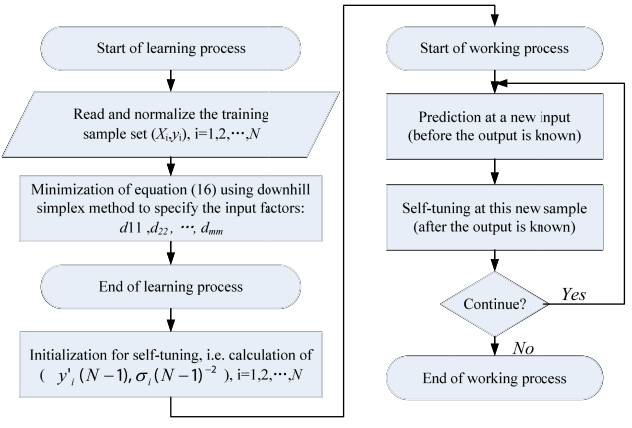


Fig. 6. Off-line training and self-tuning process of the BGNN

Note that the self-tuning of the BGNN is conducted in a sequential way which actually follows the dynamics changes. In the design of the MPC controller of HGS, the self-tuning procedure can obtain the dynamic model of the HGS when operating situations change.

5 Design of the Model Predictive Controller

The objective of the predictive control strategy using neural predictors is twofold: (i) to estimate the *future output* of the plant and (ii) to minimize a *cost function* based on the error between the predicted output of the processes and the reference trajectory. The cost function, which may be different from case to case, is minimized in order to obtain the optimum control input that is applied to the non-linear plant. In most of the predictive control algorithms a quadratic form is utilized for the cost function:

$$J = \sum_{i=N_1}^{N_2} [y(k+i) - r(k+1)]^2 + \lambda \sum_{i=1}^{N_2} u^2(k+i-1) \tag{21}$$

with additional requirements:

$$\Delta u(k+i-1)=0, \qquad 1 \le N_u < i \le N_2 \tag{22}$$

where N_u is the control horizon, N_1 is the minimum prediction horizon, N_2 is the prediction horizon, i denotes the order of the predictor, r is the reference trajectory, λ is the weight factor and Δ denotes the differentiation operator.

The command u may be subject to amplitude constraints:

$$u_{\min} \le u(k+i) \le u_{\max} \tag{23}$$

The cost function is often used with the weight factor λ=0. A very important parameter in predictive control strategy is the control horizon N_u, which specifies the instant time, since when the output of the controller should be kept at a constant value.

The output sequence of the optimal controller is obtained over the prediction horizon by minimizing the cost function J with respect to the vector u. This can be achieved by setting

$$\frac{\partial J}{\partial u}=0, \qquad u=[u(k-d),u(k-d+1),\cdots,u(k-d+N_u-1)]^T \tag{24}$$

However, a major inconvenience occurs when proceeding further with the calculation of $\partial J/\partial u$. The analytical approach to the optimization problem needs for the differentiation of the cost function and, finally, leads to a non-linear algebraic equation; unfortunately this equation cannot be solved by any analytic procedure. This is why a computational method is preferred for the minimization of the cost function, also complying with the typical requirements of the real-time implementations (guaranteed convergence, at least to a sub-optimal solution, within a given time interval).

The computation of the optimal control signal at the discrete time instant k can be achieved with the following algorithm:

• The minimization procedure performed at the previous time instant gives the command vector:

$$u=[u(k-d-1),u(k-d),\cdots,u(k-d+N_u-2)]^T \tag{25}$$

At the first time instant, the control input vector will contain some initial values provided by the user.

The number of values introduced must be equal to the control horizon.

• The step ahead predictors of orders between N_1 and N_2 are calculated by using the vectors $u(k-d+N_1-1)$, $y(k+N_1-1)$ and $u(k-d+N_2-1)$, $y(k+N_2-1)$, respectively, calculated through the BGNN predictive model described in section 4.

• The output control signal is obtained by minimizing the cost function J with respect to the command vector:

$$u=[u(k-d),u(k-d+1),\cdots,u(k-d+N_u-1)]^T \tag{26}$$

Thus, two routines were developed to implement the control algorithm in the *Matlab* environment. The main routine (*main_unc*—version without constraints, *main con*—version with constraints) minimizes the cost function and applies the command to the non-linear system. It also creates two vectors in which the output command and the

process response are stored. The other routine (*neur_prd*) performs the computations necessary for the BGNN predictors and constructs the cost function to be minimized by the main routine.

For the minimization of the cost function, the *Matlab*'s Optimal Toolbox functions *fminunc* and *fmincon* were used, which allow dealing with either unconstrained or constrained optimization problems. Unlike the *fminunc* function, *fmincon* allows imposing constraints with respect to the value of the control input such as upper or lower bounds, which are often required in practice. The cost function *J* is given as an input parameter for the functions mentioned above together with the initial values for the control input vector and some options regarding the minimization method (the maximum number of iterations, the minimum error value, the use of analytical gradients, etc.). In the case of *fmincon* the constraints must also be specified as input parameters in a matrix form.

The advantage of this non-linear neural predictive controller consists in the implementation method that solves the key problems of the non-linear MPC. The implementation is robust, easy to use and fulfills the requirements imposed for the minimization algorithm. Changes in the parameters of the BGNN predictive controller (such as the prediction horizons, the control horizon, as well as the necessary constraints) are straightforward operations.

6 Simulations and Results

The MPC control strategy based on BGNN is to be applied to a waterpower plant. The type of hydraulic turbine system is HL220 with rotating rating 300r/min, water high 33m, water flux rating 58 m^3/s and power rating 17.5 WM. The water flux inertia time constant is 3.5s and generator inertia time constant is 8.5s.

The traditional PID controller of HGS is used to be compared with the BGNN predictive controller. The dynamic response of 2Hz frequency step change is shown in Fig. 7. From the figure, it can be seen that the BGNN MPC controller can trace the step change smoothly with no oscillation while the PID controller has about 6% overshoot. The 30% load change then is conducted and the response result of BGNN MPC controller and PID controller is shown in Fig.8. It also can be seen that BGNN MPC controller owns the characteristic of steady and fast response.

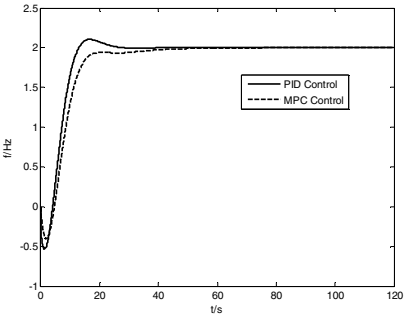


Fig. 7. Dynamic response of 2Hz frequency change

When the frequency 2Hz change and 30% load change are applied to the HGS, the BGNN identification nonlinear model of HGS is shown in Fig.9 and Fig.10. From the figure it can be seen that the accuracy of the identified HGS model can be maintained. Therefore the BGNN model can be employed for the MPC controller design.

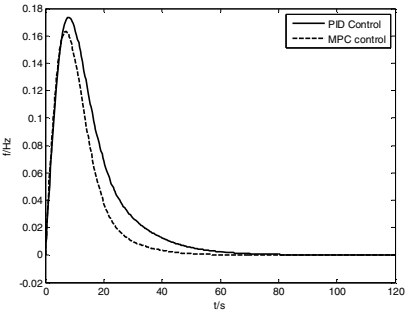


Fig. 8. Dynamic response of 30% load change

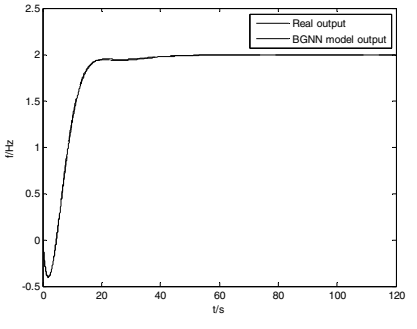


Fig. 9. Identified BGNN model output with 2Hz frequency change

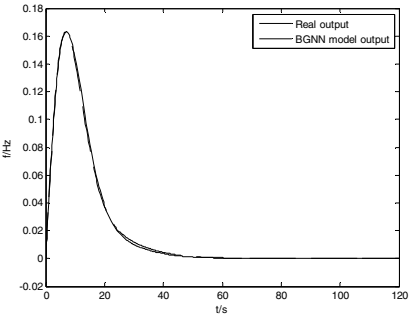


Fig. 10. Identified BGNN model output with 30% load change

7 Conclusion

A MPC strategy based on BGNN predictive model for HGS is developed in this paper. The characteristics of HGS often changes along with its operating situations. The BGNN neural network can trace the dynamic characteristics well. So the BGNN is employed as the predictive model for the design of MPC controller. Simulations on the HGS control compared with traditional PID controller shows that the proposed MPC control strategy works well and can be used to the adaptive control of HGS. The next work of this paper will be focus on the comparisons between BGNN MPC control strategy and other often used HGS intelligent control strategies such as fuzzy control strategy and neural network control strategy.

Acknowledgments. This work was supported by the National Science Foundation of China (No.60704024, No.60772107).

References

1. Jiang, C.: Nonlinear Simulation of Hydro turbine Governing System Based on Neural Network. In: IEEE International Conference on System, Man and Cybernetics, pp. 784–787 (1996)
2. Xu, F., Li, Z.: Computer Simulation about Hydraulic Generator Set. Hydraulic and Electric Power Press, Beijing (1998)
3. Prakash, J., Senthil, R.: Design of Observer Based nonlinear Model Predictive Controller for a Continuous Stirred Tank Reactor. Journal of Process Control 18, 504–514 (2008)
4. Bellemans, T., De Schutter, B., De Moor, B.: Model Predictive Control for Ramp Metering of Motorway Traffic: a Case Study. Control Engineering Practice 14(5), 441–466 (2006)
5. Blasco, X., Martinez, M., et al.: Model-Based Predictive Control of Greenhouse Climate for Reducing Energy and Water Consumption. Computer and Electronics in Agriculture 55(1), 49–70 (2007)
6. Funahashi, K.: On the approximation of continuous mapping by neural networks. Neural Netw. 2, 183–192 (1989)
7. Yazdan, S., Mohsen, H., Rostam, M.: Numerical Solution of the Nonlinear Schrodinger Equation by Feedforward Neural Networks. Communications in Nonlinear Science and Numerical Simulation 13(10), 2132–2145 (2008)
8. Jiang, C., Xiao, Z., Wang, S.: Neural Network Predict Control for the Hydro Turbine Generator Set. In: The Second International Conference on Machine Learning and Cybernetics, pp. 2–5 (2003)
9. Ye, H., Nicolai, R., Reh, L.: A Bayesian-Gaussian Neural Network and Its Application in Process Engineering. Chemical Engineering and Process 38, 439–449 (1998)
10. Ye, H., Ni, W.: Nonliner System Identification Using a Bayesian-Gaussian Neural Network for Predictive Control. Neurocomputing 28, 21–36 (1999)

The Application of Improved BP Neural Network Algorithm in Lithology Recognition

Yuxiang Shao[1], Qing Chen[2], and Dongmei Zhang[1]

[1] China University of Geosciences, No.388 Lumo Road Wuhan, P.R. China, China
syxcq@163.com
[2] Wuhan Institute of Technology, Wuhan, Hubei, P.R. China, China
chenqing@mail.wit.edu.cn

Abstract. Traditional technology of lithology identification bases on statistical theory, such as regression method and cluster method, which has some shortcomings. The standard BP neural network algorithm has some disadvantages like slow convergence speed, local minimum value which results in the loss of global optimal solution. BP neural network algorithm on the basis of improved variable rate of momentum factor can effectively overcome these disadvantages. Practical application shows that this method has the feature as high recognition precision and fast recognition rate so that it is suitable for recognition of lithology, lithofacies and sedimentary facies as well as geological research like deposit prediction and rock and mineral recognition.

Keywords: Lithology Recognition, Improved BP Algorithm, Logging curves, Momentum factor.

1 Introduction

Logging parameters is a comprehensive reflection of mineral compositions, contracture and porosity of underground rocks[1]. A set of certain logging curves must correspond to one or several lithology[2]. Based on the analysis of eigenvalues corresponding to rock cores and logging parameters in study area, the method classifies the rock types and get representative logging parameters from each type of rock to serve as basic data for determining the relation between lithology and logging parameters, and then establish lithology identification mode by mathematical means[3]. That is the basic cross plot method and numerical simulation method for geological interpretation of logging data[4]. However, those two methods have some limitations.

BP neural network algorithm, as one of data mining technologies, has good capability of parallel architecture and parallel processing with the inherent nonlinear characteristics and self-learning, self-organization and self-adapting capabilities[5~7]. It can automatically attain knowledge by example training without analysis and arrangement and conduct effectively modeling to those information and technologies which cannot be used to establish accurate model by mathematical method[8]. However, as BP neural network learns on the basis of error back propagation algorithm of gradient descent, it has slow network training speed and tends to yield minimal local results[9~10].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 342–349, 2008.
© Springer-Verlag Berlin Heidelberg 2008

This study applies BP neural network algorithm to lithology recognition. Improved BP neural network algorithm based on momentum factor is adopted to the logging curve of the 11th well in Saihantala Depression. The result shows that improved BP neural network model has high prediction precision and good generalization ability so that it is effective for lithology prediction.

2 BP Neural Network Algorithm

BP neural network algorithm includes the following parts:

1. Input N training examples $\left(x_k , y_k^* \right)$, here x_k represents input vector, y_k^* represents expected output vectors, k=l, 2, ···, N。

2. Establish BP neural network. The node number of input layer in network determined by the length n of input vector x_k is n, and the node number of output layer determined by the length m of output vector y_k^* is m. Determine the node number with network layer L≥3, the node number of the l layer is $n^{(l)}$ and $n^{(l)}$=n, $n^{(L)}$=m. Define each connection weight matrix, the connection matrix in the L layer connecting L+1 layer is $W^{(l)} = \left[w_{ij}^{(l)} \right]_n^{(l)} \times_n^{(l+1)}, l = 1,2,\cdots, L-1$, initialize the values of each connection weight matrix.

3. Input permissible error ε and learning rate η, initialize the times of iteration t=l, number of training examples k=l。

4. Choose the k th training example (x_k , y_k^*):

$$x_k = \left(x_{1k}, x_{2k}, \cdots, x_{nk} \right), y_k^* = \left(y_{1k}^*, y_{2k}^*, \cdots y_{mk}^* \right)$$

5. Forward propagation is calculated from x_k and output of each node in input layer is:

$$O_{jk}^{(l)} = f(x_{jk}), j = 1,2,\cdots, n \tag{1}$$

The input and output of each node is calculated layer by layer:

$$I_{jk}^{(l)} = \sum_{i=1}^{n^{(l-1)}} w_{ij}^{(l-1)} O_{ik}^{(l-1)} \tag{2}$$

Among them, $O_{jk}^{(l)} = f\left(I_{jk}^{l} \right), l = 2,\cdots, L; j = 1,2,\cdots, n^{(l)}$

6. Error of each output node in the Lth output layer is calculated by:

$$y_{jk} = O_{jk}^{(L)} \tag{3}$$

$$E_{jk} = \frac{1}{2} e_{jk}^2 = \left(y_{jk}^* - y_{jk} \right)^2 \Big/ 2, j = 1,2,\cdots,m \tag{4}$$

7. If the K value of any sample in N training examples $E_{jk} \leq \varepsilon$, j=1, 2, …, m, the training process terminates; Otherwise, each connection weight matrix is corrected by error back propagation.

8. Error back propagation calculation. The connection weight matrix of the L-1th hidden layer to the Lth output layer is corrected as:

$$\delta_{jk}^{(L)} = -\left(y_{jk}^* - y_{jk} \right) f'\left(I_{jk}^{(L)} \right) \tag{5}$$

$$\Delta w_{ij}^{(L-1)}(t) = \eta \delta_{jk}^{(L+1)} O_{ik}^{(L)} \tag{6}$$

$$w_{ij}^{(L-1)}(t+1) = w_{ij}^{(L-1)}(t) + \Delta w_{ij}^{(L-1)}(t) \tag{7}$$

Among them, j=1, 2, …, m ; i=1, 2, …, $n^{(L-1)}$

The connection weight matrix of each hidden layer is corrected backward layer by layer:

$$\delta_{jk}^{(l)} = f'\left(I_{jk}^{(l)} \right) \sum_{q=1}^{n^{(l+1)}} \delta_{qk}^{(l+1)} w_{jq}^{(l)} \tag{8}$$

$$\Delta w_{ij}^{(l-1)}(t) = -\eta \delta_{jk}^{(l)} O_{ik}^{(l-1)} \tag{9}$$

$$w_{ij}^{(l-1)}(t+1) = w_{ij}^{(l-1)}(t) + \Delta w_{ij}^{(l-1)}(t) \tag{10}$$

Among them, l=L-1,…,2,1;j=1,2,…,$n^{(l)}$;i=1,2,…, $n^{(l-1)}$

9. If k=k+l, t=t+1, then turn to step 4.

3 Algorithm Improvement

According to the shortcomings of standard BP neural network algorithm, such as slow convergence speed, local minimum value which results in the loss of global optimal solution, sensitive to the choose of initial weight, bad generalization ability and so on, BP neural network algorithm is improved. The way of improvement is to add the momentum factor and change the learning rate and this improvement mainly aims at the above formula (9).

1. The adding of momentum factor

As weight change is needed during the learning process of BP neural network, results with much differences might be gained or seriously oscillate around the minimum value for bigger learning factorη. In order to make some remedy to the oscillation, a momentum factor is introduced into formula (9) which depends on the increment value of weight parameter in the last iteration.

$$\Delta w_{ij}^{(l-1)}(t) = \alpha \Delta w_{ij}^{(l-1)}(t-1) - \eta \delta_{jk}^{(l)} O_{ik}^{(l-1)} \tag{11}$$

In the formula, α is momentum factor, usually positive.

As $\Delta w_{ij}^{(l-1)}(t)$ is the sum of a series weighted index sequence, the sequence convergent when 0≤α<1.

As $E_k = \frac{1}{2}\sum_{j=1}^{m} e_{jk}^2$, when $\frac{\partial E_k}{\partial w_{ij}}$ in this time has the same symbol as the last time, the exponential weighting addition increases to attain bigger $\Delta w_{ij}^{(l-1)}(t)$ so as to speed the adjustment of weight value w during the steady regulating.

When $\frac{\partial E_k}{\partial w_{ij}}$ in this time has the opposite symbol as the last time, the result of exponential weighting addition $\Delta w_{ij}^{(l-1)}(t)$ decreases and makes it stable.

2. The changing of learning rate

η is a constant in standard BP neural network algorithm. The changing of learning rate can accelerate the learning process. The effective algorithm is: total error of all samples E is calculated on the basis of last step length. If E increases, this step length will multiply a constant β smaller than 1 and E of the next iteration point will be recalculated along primary direction; if E decreases after one time iteration, then this step length will multiply a constant φ bigger than 1.

That is : $\eta = \eta\beta, \beta < 1,$ if $\Delta E > 0$

$\eta = \eta\varphi, \varphi > 1$, if $\Delta E < 0$

β, φ are constants, usually β is 0.7, φ is 1.05.

The adding of momentum factor in the BP neural network algorithm changes the learning rate, not only fine adjusting the correction value of weight, but also avoiding getting minimal local results so as to reach self-adaptation.

4 Application Example

4.1 Examples and Data

In this study, logging data of the key well S11 in Saihantala Depression is used to conduct algorithm training and 30 glutenite samples, 30 shale samples and 30 silt samples are chosen as training samples. According to principal component analysis, there are 6 variables, R045, RA05, R4, SP, AC, GR, used as input attributes and in the result, (1, 0, 0), (0, 1, 0) and (0, 0, 1) stand for glutenite, shale and silt, respectively (only selected training samples are shown in table 1).

4.2 Normalization Processing

As all the logging data have inconsistent dimensions, normalization processing is needed for both learning samples and prediction data to unify them into a certain range of numerical dimension (e.g. [0, 1]) before entering the neural network.

Table 1. Training cases of prediction model

R045	RA05	R4	SP	AC	GR	lithology (code)
8.030	11.154	9.947	17.126	550.223	5.919	glutenite (1, 0, 0)
8.891	11.612	10.296	17.125	488.011	6.125	glutenite (1, 0, 0)
8.773	11.042	8.993	18.042	462.863	6.663	glutenite (1, 0, 0)
8.236	11.016	8.718	17.913	468.876	6.508	glutenite (1, 0, 0)
7.756	10.990	8.510	17.812	473.889	6.224	glutenite (1, 0, 0)
3.300	4.381	4.834	20.717	615.072	7.379	shale (0, 1, 0)
3.280	4.420	4.748	20.716	614.082	7.384	shale (0, 1, 0)
3.320	4.396	5.047	20.618	615.054	7.488	shale (0, 1, 0)
3.320	4.381	4.961	20.617	615.063	7.359	shale (0, 1, 0)
3.259	4.422	4.621	20.715	610.435	7.438	shale (0, 1, 0)
5.139	6.349	7.261	19.008	475.497	7.579	silt (0, 0, 1)
5.152	6.472	7.072	18.792	470.511	7.588	silt (0, 0, 1)
5.272	7.051	6.549	18.175	444.183	7.112	silt (0, 0, 1)
5.227	7.430	6.602	18.074	455.125	7.035	silt (0, 0, 1)
5.200	7.695	6.710	17.973	467.481	6.958	silt (0, 0, 1)

$$x_{ij}^{'} = \frac{(x_{j\max} - x_{ij})}{(x_{j\max} - x_{j\min})}, (i = 1,2,\cdots, sample \quad number; j = 1,2,\cdots,6)$$

Among them, $x_{j\max}$ =max{ x_{ij}}, $x_{j\min}$ =min{ x_{ij}} is maximum value and minimum value of each variable.

Normalized formula with logarithmic method is adopted to curves with nonlinear characteristics like electrical resistivity.

$$x_{ij}^{'} = \frac{(\lg x_{j\max} - \lg x_{ij})}{(\lg x_{j\max} - \lg x_{j\min})}$$

4.3 Implementation and Application of Improved BP Neural Network

Three-layer BP neural network (one hidden layer) is applied and there are 6 neurons in input layer, 10 neurons in hidden layer and 3 neurons in output layer. Initial training parameters are: learning step η is 1.02, scaling factor of increasing learning rate φ is 1.05, scaling factor of decreasing learning rate β is 0.7, momentum factor α is 0.92, allowable error (MSE) is 0.001. Network convergence is greatly accelerated using improved BP neural network.

5 glutenite samples, 5 shale samples and 5 silt samples which are not used for model establishment in the S11 logging data are adopted as prediction set for testing the logging lithology recognition. The result shows that, except for little error in a few samples, others have complete consistency in expected output values and the

Table 2. Predicted results

R045	RA05	R4	SP	AC	GR	Petrologic Features			BP algorithm Preditcion		
8.994	9.147	10.400	15.933	453.847	6.419	**1**	**0**	**0**	0.8874	0.1385	0.0000
9.449	9.455	10.896	16.037	460.078	6.260	**1**	**0**	**0**	0.9996	0.0011	0.0000
9.753	9.396	10.917	16.133	466.233	6.211	**1**	**0**	**0**	0.9999	0.0000	0.0001
9.144	9.271	10.338	16.132	473.390	6.412	**1**	**0**	**0**	0.9997	0.0002	0.0004
8.654	9.155	9.851	16.172	480.085	6.629	**1**	**0**	**0**	1.0000	0.0001	0.0000
3.304	4.613	6.097	20.500	569.384	7.870	**0**	**1**	**0**	0.0000	1.0000	0.0000
3.238	4.573	5.976	20.499	557.525	7.880	**0**	**1**	**0**	0.0000	1.0000	0.0000
3.218	4.572	5.947	20.498	544.273	7.837	**0**	**1**	**0**	0.0001	1.0000	0.0000
3.288	4.645	5.946	20.497	531.219	7.729	**0**	**1**	**0**	0.0000	1.0000	0.0002
3.355	4.888	6.013	20.396	520.156	7.470	**0**	**1**	**0**	0.0000	0.9999	0.0001
5.252	5.943	6.401	18.404	467.417	7.370	**0**	**0**	**1**	0.0000	0.0013	0.9994
5.161	6.046	6.201	18.504	472.540	7.429	**0**	**0**	**1**	0.0000	0.0012	0.9995
5.039	6.112	6.024	18.603	475.716	7.304	**0**	**0**	**1**	0.0000	0.0005	0.9998
5.139	6.125	6.185	18.804	478.799	7.305	**0**	**0**	**1**	0.0000	0.0000	0.9999
5.274	6.009	6.508	18.989	483.234	7.388	**0**	**0**	**1**	0.0000	0.0001	1.0000

lithology recognition is completely correct. MSE value of predicted samples is 0.0012. Predicted results are shown in table 2.

5 Comparative Analysis of Algorithm

The standard BP neural network algorithm and improved BP neural network algorithm are applied to well logging data above for training and forecasting. The comparative result of algorithm for model design and training times is shown in table 3.

Table 3. Algorithm comparison

Algorithm	main parameter	training times
standard BP algorithm	η=1.02	41633
improved BP algorithm	η=1.02 α=0.92 φ=1.05 β=0.7	2531

The diagram of training and prediction errors calculated by standard BP algorithm is shown in figure 1 and the diagram of training and prediction errors calculated by improved BP algorithm is shown in figure 2.

The result indicates that the standard BP neural network algorithm has shortcomings of slow convergence rate and proneness to yield minimal local results which results in training termination by reaching minimum gradient （generally 1e-10） so that training error cannot be attained. Besides, the standard BP algorithm has bad

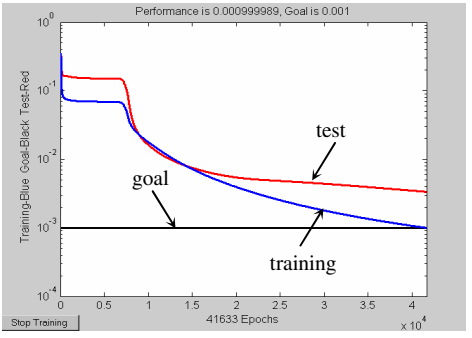


Fig. 1. Diagram of error curves for standard BP algorithm

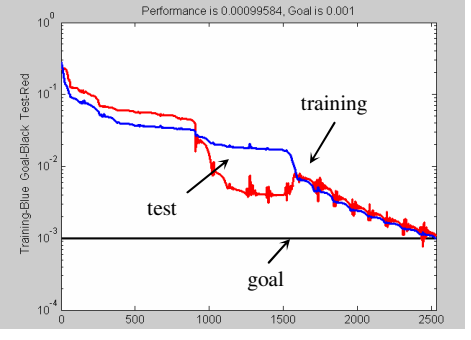


Fig. 2. Diagram of error curves for improved BP algorithm

generalization ability and large error of prediction data. Improved BP neural network algorithm not only speed the training process, but also ensure the training stability when conducting weight regulation. It corrects the shortcomings of standard BP algorithm and is developed for practical application.

6 Conclusion

The study on improved BP neural network algorithm based on momentum factor is supported by petroleum exploration project in Saihantala Depression, Erlian Basin. This algorithm is not only applied to model testing in the key well S11, but also to lithology prediction in other key wells. The research indicates that, the establishment of lithology prediction model using 6 logging curves R045、RA05、R4、SP、AC、GR can greatly improve the accuracy of lithology recognition. According to the limitations of traditional BP algorithm, momentum factor is added to adjust the learning rate and speed the convergence of neural network. Improved algorithm model has certain representation and good simulating and predicting abilities and its accuracy can meet the prediction demand. The method of improved BP neural network algorithm applied to lithology recognition is simple, accurate with a good reproducibility

and can be used as a brand new method for geologic interpretation of logging data so that it has practical significance in recognizing hydrocarbon-bearing formation as well as exploring oil and gas resources.

References

1. Cheng, G., Zhou, G., Wang, X.: The Probability Neural Networks for Lithology Identification. Control & Automation 23(6), 288–290 (2007)
2. Chen, R.: Research on the Application of BP Neural Networks. Control & Automation 23(8), 258–259 (2007)
3. Zhou, G., Cheng, G.: BRF neural networks for petroleum reservoir characterization. Computer Engineering and Applications 43(8), 174–176 (2007)
4. Zhou, B., Li, Z., Pan, B.: A Study on Lithology Identification Methods for Volcanic Rocks. Journal of Jilin University (Earth Science Edition) 35(3), 394–397 (2005)
5. Zhang, H., Zou, L., Shen, X.: The Application of BP Neural Network in Well Lithology Identification. Geology and Prospection 38(6), 63–65 (2002)
6. Xie, L.: BP Neural Network Algorithm Improvements and Convergence Analysis. Computing Technology and Automation 26(3), 52–56 (2007)
7. Hopfield, J.J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. Nat. Acad. Sci. (79), 554–558 (1982)
8. Dony, R.D., Simon: Neural Network Approaches to Image Compression. Proceedings of the IEEE 83, 288–303 (1995)
9. Fritzke, B.: Growing cell structure – A self-organizing neural network for unsupervised and supervised learning. Neural Networks 7, 441–460 (1994)
10. Yarlagadda, P.K.D.V., Chiang, E.C.W.: A neural network system for the prediction of process parameters in pressure die casting. Mater. Proc. Technol. 89-90, 583–590 (1999)

A Combined Classification Algorithm Based on C4.5 and NB

Liangxiao Jiang[1], Chaoqun Li[2], Jia Wu[1], and Jian Zhu[1]

[1] Faculty of Computer Science, China University of Geosciences
Wuhan, Hubei, P.R. China, 430074
ljiang@cug.edu.cn
[2] Faculty of Mathematics, China University of Geosciences
Wuhan, Hubei, P.R. China, 430074
chqli@cug.edu.cn

Abstract. When our learning task is to build a model with accurate classification, C4.5 and NB are two very important algorithms for achieving this task because of their simplicity and high performance. In this paper, we present a combined classification algorithm based on C4.5 and NB, simply C4.5-NB. In C4.5-NB, the class probability estimates of C4.5 and NB are weighted according to their classification accuracy on the training data. We experimentally tested C4.5-NB in Weka system using the whole 36 UCI data sets selected by Weka, and compared it with C4.5 and NB. The experimental results show that C4.5-NB significantly outperforms C4.5 and NB in terms of classification accuracy. Besides, we also observe the ranking performance of C4.5-NB in terms of AUC (the area under the Receiver Operating Characteristics curve). Fortunately, C4.5-NB also significantly outperforms C4.5 and NB.

Keywords: decision trees, naive Bayes, combined algorithms, weights, classification, ranking, data mining.

1 Introduction and Related Work

Classification is one of the fundamental problems in data mining and knowledge discovery. In classification, the goal is to learn a classifier from a given set of instances with class labels, which correctly assigns a class label to a test instance. The performance of a classifier is usually measured by its classification accuracy (the percentage of instances correctly classified). Classification has been extensively studied and various learning algorithms have be developed, such as decision tree and Bayesian network, that can be categorized into two major approaches: probability-based approach and decision boundary-based approach. In this paper, we focus on the probability-based approach.

Decision tree [1] is one of the most widely used classification models. It classifies a test instance by sorting it down the tree from the root node to one leaf node, which provides the classification of this instance via simply voting. Each node in the tree specifies a test of one attribute of the instance, and each

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 350–359, 2008.
© Springer-Verlag Berlin Heidelberg 2008

branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given instance. This process is then repeated for the subtree rooted at the new node. After the tree is built, a manipulation called tree pruning is performed to scale up the classification accuracy of the learned tree. Among decision tree classification algorithms, C4.5 [2] is the most popular one, which is the improved version of ID3 [3].

A Bayesian network [4] consists of a structural model and a set of conditional probabilities. The structural model is a directed acyclic graph in which nodes represent attributes and arcs represent attribute dependencies. Attribute dependencies are quantified by conditional probabilities for each node given its parents. The Bayesian network is often used for classification problems, in which a learner attempts to construct a Bayesian network classifier [6] from a given set of training instances with class labels. Assume that all attributes are independent given the class, then the resulting Bayesian network classifier is called the naive Bayesian classifier, simply NB [5].

Assume that A_1, A_2, $\cdots$, A_m are m attributes. A test instance x is represented by a vector $< a_1, a_2, \cdots, a_m >$, where a_i is the value of A_i. We use C to denote the class variable and c to denote its value, and $c(x)$ to denote the class of the test instance x. In a probability-based learning algorithm, a joint probability distribution $P(a_1, a_2, \cdots, a_m, c)$ is learned from the training data, and x is classified into the class $c(x)$ with the maximum posterior class probability $P(c|a_1, a_2, \cdots, a_m)$ (or simply, class probability), as shown below:

$$c(x) = \arg\max_{c \in C} P(c|x) = \arg\max_{c \in C} P(c|a_1, a_2, \cdots, a_m) \quad (1)$$

In this paper, we present a combined classification algorithm based on C4.5 and NB, simply C4.5-NB. In C4.5-NB, the class probability estimates of C4.5 and NB are weighted according to their classification accuracy on the training data. We experimentally tested C4.5-NB in Weka system using the whole 36 UCI data sets [7] selected by Weka [8], and compared it with C4.5 and NB. The experimental results show that C4.5-NB significantly outperforms C4.5 and NB in terms of classification accuracy. Due to its simplicity and high performance, C4.5-NB should be widely used for addressing classification problems in real-world data mining and knowledge discovery applications.

The rest of the paper is organized as follows. In Section 2, we present our combined algorithm, simply called C4.5-NB. In Section 3, we describe the experimental setup and results in detail. In Section 4, we draw conclusions and outline our main directions for future work.

2 A Combined Classification Algorithm Based on C4.5 and NB: C4.5-NB

In this section, we present our combined algorithm, simply called C4.5-NB. Our motivation is to scale up the classification accuracy of C4.5 and NB. In

C4.5-NB, the class probability estimates of C4.5 and NB are weighted according to their classification accuracy on the training data. Now, let's look back the class probability estimates of C4.5 and NB.

C4.5 estimates the probability of a test instance x belonging to class c using simple voting at the leaf where x falls into. The detailed equation is:

$$P(c|x)_{C4.5} = \frac{\sum_{i=1}^{k} \delta(c_i, c)}{k} \tag{2}$$

where k is the number of training instances in the leaf node L, c_i is class label of the ith training instance in the leaf node L, and $\delta(c_i, c)$ is a binary function, which is one if $c_i = c$ and zero otherwise.

NB is the most simple Bayesian network classifers, which uses Equation 3 to estimate the probability of x belonging to class c.

$$P(c|x)_{NB} = P(c) \prod_{j=1}^{m} P(a_j|c) \tag{3}$$

where m is the number of attributes, a_j is the jth attribute value of x, the prior probability $P(c)$ and the conditional probability $P(a_j|c)$ is respectively defined using Equation 4 and Equation 5.

$$P(c) = \frac{\sum_{i=1}^{n} \delta(c_i, c) + 1}{n + n_c} \tag{4}$$

$$P(a_j|c) = \frac{\sum_{i=1}^{n} \delta(a_{ij}, a_j)\delta(c_i, c) + 1}{\sum_{i=1}^{n} \delta(c_i, c) + n_j} \tag{5}$$

where n is the number of training instances, n_c is the number of classes, n_j is the number of values of the jth attribute, and a_{ij} is the jth attribute value of the ith training instance.

Our C4.5-NB algorithm is an ensemble algorithm, which combines C4.5 and NB. At the training time, C4.5-NB simultaneity trains C4.5 and NB, and measures their classification accuracy (respectively denoted by $ACC_{C4.5}$ and ACC_{NB}) on the training data. At the test time, the class probability estimates of C4.5 and NB are weighted according to $ACC_{C4.5}$ and ACC_{NB}. The detailed training and test algorithms can be depicted as:

Algorithm. Training (**D**)
Input: the training data **D**
Output: the C4.5 and NB classifiers, $ACC_{C4.5}$, and ACC_{NB}

1. Builds a C4.5 classifier using **D**
2. Measures $ACC_{C4.5}$ of the built C4.5 classifier on **D**
3. Builds an NB classifier using **D**
4. Measures ACC_{NB} of the built NB classifier on **D**
5. Returns the built C4.5 and NB classifiers, $ACC_{C4.5}$, and ACC_{NB}

Algorithm. Test (C4.5, NB, $ACC_{C4.5}$, ACC_{NB}, **x**)
Input: the built C4.5 and NB, $ACC_{C4.5}$, ACC_{NB}, and a test instance **x**
Output: the class label of **x**

1. For each possible class label c
2. Uses C4.5 to estimate the probability $P(c|x)_{C4.5}$ via Equation 2
3. Uses NB to estimate the probability $P(c|x)_{NB}$ via Equation 3
4. Calculates $P(c|x)$ via $\frac{ACC_{C4.5} \times P(c|x)_{C4.5} + ACC_{NB} \times P(c|x)_{NB}}{ACC_{C4.5} + ACC_{NB}}$
5. Returns the class c with maximal $P(c|x)$ value as the class label of **x**

The experimental results in Section 3 show that C4.5-NB significantly outperforms C4.5 and NB in terms of classification accuracy. Besides, C4.5-NB also has relatively low time complexity thanks to the very low time complexity of C4.5 and NB. If we use $O(C4.5)$ and $O(NB)$ to denote the time complexity of C4.5 and NB, then the time complexity of C4.5-NB can be denoted by $O(C4.5)+O(NB)+O(2n)$, where the third item $O(2n)$ is the time complexity of calculating the weights of the C4.5 and NB classifiers. In a word, we believe that our new algorithm C4.5-NB will be widely used for classification in real-world data mining and knowledge discovery applications.

3 Experimental Methodology and Results

In order to investigate the classification performance of our new algorithm C4.5-NB, we design an empirical study in this section. In our experiments, 36 UCI data sets [7] are used:

1. In order to avoid the trouble of choosing data sets, we directly use the whole 36 UCI data sets published by main web site of Weka [8]. Just as we all know, there exist a lot of data sets in the UCI repository of machine learning databases [7]. It is very easy to make our advantage stronger, if we add some other data sets into our experiments.
2. To our knowledge, these 36 UCI data sets have already represented a wide range of domains and data characteristics. The detailed data information can be found in Table 1.
3. We did all experiments in Weka software. So, all data set must be in format of arff. We can directly download these data sets in format of arff from main web site of Weka [8]. If we use some other data sets, we must take a lot of time to convert the format of data stes.
4. In fact, many authors, for example, Zhang [9] and Liang [10], use these data sets for testing their algorithms.

Besides, we adopted the following three preprocessing steps.

1. Replacing missing attribute values: Due to simplicity reasons, we don't handle missing attribute values. Thus, we used the unsupervised filter named *ReplaceMissingValues* in Weka to replace all missing attribute values in each data set.

Table 1. Description of data sets used in the experiments. All these data sets are the whole 36 UCI data sets selected by Weka. We downloaded these data sets in format of *arff* from main web site of weka.

No.	Data set	Instances	Attributes	Classes	Missing	Numeric
1	anneal	898	39	6	Y	Y
2	anneal.ORIG	898	39	6	Y	Y
3	audiology	226	70	24	Y	N
4	autos	205	26	7	Y	Y
5	balance-scale	625	5	3	N	Y
6	breast-cancer	286	10	2	Y	N
7	breast-w	699	10	2	Y	N
8	colic	368	23	2	Y	Y
9	colic.ORIG	368	28	2	Y	Y
10	credit-a	690	16	2	Y	Y
11	credit-g	1000	21	2	N	Y
12	diabetes	768	9	2	N	Y
13	Glass	214	10	7	N	Y
14	heart-c	303	14	5	Y	Y
15	heart-h	294	14	5	Y	Y
16	heart-statlog	270	14	2	N	Y
17	hepatitis	155	20	2	Y	Y
18	hypothyroid	3772	30	4	Y	Y
19	ionosphere	351	35	2	N	Y
20	iris	150	5	3	N	Y
21	kr-vs-kp	3196	37	2	N	N
22	labor	57	17	2	Y	Y
23	letter	20000	17	26	N	Y
24	lymph	148	19	4	N	Y
25	mushroom	8124	23	2	Y	N
26	primary-tumor	339	18	21	Y	N
27	segment	2310	20	7	N	Y
28	sick	3772	30	2	Y	Y
29	sonar	208	61	2	N	Y
30	soybean	683	36	19	Y	N
31	splice	3190	62	3	N	N
32	vehicle	846	19	4	N	Y
33	vote	435	17	2	Y	N
34	vowel	990	14	11	N	Y
35	waveform-5000	5000	41	3	N	Y
36	zoo	101	18	7	N	Y

2. Discretizing numeric attribute values: Due to simplicity reasons, we don't handle numeric attribute values. Thus, we used the unsupervised filter named *Discretize* in Weka to discretize all numeric attribute values in each data set.
3. Removing useless attributes: Apparently, if the number of values of an attribute is almost equal to the number of instances in a data set, it is a useless attribute. Thus, we used the unsupervised filter named *Remove* in Weka to remove this type of attributes. In these 36 data sets, there are only three such

attributes: the attribute "Hospital Number" in the data set "colic.ORIG", the attribute "instance name" in the data set "splice" and the attribute "animal" in the data set "zoo".

We implemented our C4.5-NB within the Weka framework and used the implementation of naive Bayes (NiaveBayes) and C4.5 (J48) in Weka. In all experiments, the classification accuracy and standard deviation of an algorithm on a

Table 2. Experimental results on classification accuracy and standard deviation. v, *: statistically significant improvement or degradation over C4.5-NB.

Dataset	C4.5-NB	C4.5	NB
anneal	98.65±0.97	98.65±0.97	94.32±2.23 *
anneal.ORIG	92.53±2.35	90.36±2.51 *	88.16±3.06 *
audiology	76.14±6.76	77.22±7.69	71.4±6.37
autos	80.2±9.27	81.54±8.32	63.97±11.35 *
balance-scale	74.62±3.94	64.14±4.16 *	91.44±1.3 v
breast-cancer	72.29±6.77	75.26±5.04	72.94±7.71
breast-w	97.07±2.04	94.01±3.28 *	97.3±1.75
colic	82.94±6.04	84.31±6.02	78.86±6.05 *
colic.ORIG	80.81±6	80.79±5.66	74.21±7.09 *
credit-a	86.33±4.22	85.06±4.12	84.74±3.83
credit-g	74.93±3.69	72.61±3.49 *	75.93±3.87
diabetes	75.94±4.76	73.89±4.7	75.68±4.85
glass	59.54±9.13	58.14±8.48	57.69±10.07
heart-c	81.89±6.52	79.14±6.44	83.44±6.27
heart-h	82.26±6.61	80.1±7.11	83.64±5.85
heart-statlog	82.26±6.58	79.78±7.71	83.78±5.41
hepatitis	83.16±10.18	81.12±8.42	84.06±9.91
hypothyroid	93.6±0.52	93.24±0.44 *	92.79±0.73 *
ionosphere	91.31±4.49	87.47±5.17 *	90.86±4.33
iris	95.4±5.33	96±4.64	94.33±6.79
kr-vs-kp	99.44±0.37	99.44±0.37	87.79±1.91 *
labor	92.27±10.41	84.97±14.24 *	96.7±7.27
letter	82.67±0.74	81.31±0.78 *	70.09±0.93 *
lymph	83.35±8.41	78.21±9.74 *	85.97±8.88
mushroom	100±0	100±0	95.52±0.78 *
primary-tumor	44.55±6.76	41.01±6.59 *	47.2±6.02
segment	92.77±1.75	93.42±1.67	89.03±1.66 *
sick	97.54±0.77	98.16±0.68 v	96.78±0.91 *
sonar	76.92±9.98	71.09±8.4	76.35±9.94
soybean	93.75±2.51	92.63±2.72	92.2±3.23
splice	95.63±1.07	94.17±1.28 *	95.42±1.14
vehicle	69.8±3.31	70.74±3.62	61.03±3.48 *
vote	96.43±2.79	96.27±2.79	90.21±3.95 *
vowel	79.02±4.21	75.57±4.58 *	66.09±4.78 *
waveform-5000	78.9±1.82	72.64±1.81 *	79.97±1.46 v
zoo	93.1±7.35	92.61±7.33	94.37±6.79
Mean	84.39±4.68	82.64±4.75	82.34±4.78

Table 3. Experimental results: two-tailed t-test with the 95% confidence level

	C4.5-NB	C4.5	NB
C4.5-NB	-	1	2
C4.5	13	-	8
NB	14	13	-

data set was obtained via 10 runs of 10-fold cross validation. Runs with the various algorithms were carried out on the same training sets and evaluated on the same test sets. Finally, we conducted a two-tailed t-test with a 95% confidence level [11] to compare C4.5 with NB and our C4.5-NB.

Table 2 shows the classification accuracy and standard deviation of all three algorithms on each data set. The symbols v and * in the table respectively denotes statistically significant improvement or degradation over C4.5-NB with a 95% confidence level. The average values are summarized at the bottom of the table. Table 3 shows the compared results of two-tailed t-test with a 95% confidence level between each pair of algorithms. Each number in the table indicates how many data sets the algorithm at the corresponding column significantly outperforms the algorithm at the corresponding row.

From Table 2 and Table 3, we can see that: In terms of classification accuracy, C4.5 is competitive with NB and C4.5-NB significantly outperforms C4.5 and NB. Now, let's summarize the highlights as follows:

1. C4.5 is competitive with NB, C4.5 significantly outperforms NB on 13 data sets (second column, third row), and NB significantly outperforms C4.5 on 8 data sets (third column, second row). Besides, The average value of C4.5 (82.64) almost ties that of NB (82.34).
2. C4.5-NB significantly outperforms C4.5, C4.5-NB wins C4.5 on 13 data sets (first column, second row), whereas C4.5 only wins C4.5-NB on 1 data sets (second column, first row). Besides, the average value of C4.5-NB (84.39) is much higher than that of C4.5.
3. C4.5-NB significantly outperforms NB, C4.5-NB wins NB on 14 data sets (first column, third row), whereas NB only wins C4.5-NB on 2 data sets (third column, first row). Besides, the average value of C4.5-NB (84.39) is much higher than that of NB.

In our experiments, we also observe the ranking performance of C4.5-NB in terms of AUC (the area under the Receiver Operating Characteristics curve) [12,13,14,15]. Fortunately, C4.5-NB also significantly outperforms C4.5 and NB. The detailed experimental results can be found in Table 4 and Table 5. Now, let's summarize the highlights as follows:

1. C4.5-NB significantly outperforms C4.5, C4.5-NB wins C4.5 on 35 data sets (first column, second row), whereas C4.5 wins C4.5-NB on 0 data sets (second column, first row). Besides, the average value of C4.5-NB (89.85) is much higher than that of C4.5 (82.41).
2. C4.5-NB significantly outperforms NB, C4.5-NB wins NB on 11 data sets (first column, third row), whereas NB only wins C4.5-NB on 2 data sets

Table 4. Experimental results on AUC and standard deviation. v, * : statistically significant improvement or degradation over C4.5-NB.

Dataset	C4.5-NB	C4.5		NB	
anneal	96.12±1.19	85.74±4.67	*	96.1±1.18	
anneal.ORIG	94.6±4.27	86.08±4.45	*	94.26±4.22	*
audiology	71.17±0.67	69.4±0.76	*	71.08±0.64	
autos	94.41±2.56	84.84±6.76	*	90.07±4.93	*
balance-scale	69.34±6.62	53.47±7.33	*	84.08±4.42	v
breast-cancer	68.06±11.92	56.55±10.07	*	68.22±11.94	
breast-w	98.86±1.01	96.32±3.11	*	99.22±0.76	
colic	85.57±7.59	80.95±8.07	*	83.22±7.46	
colic.ORIG	87±5.7	83.02±7.63	*	80.57±8.03	*
credit-a	91.96±3.22	88.43±3.93	*	91.71±3.16	
credit-g	77.96±4.04	68.94±5.49	*	79.02±4.22	
diabetes	82.89±5.76	76.26±6.94	*	82.51±5	
glass	85.21±4.25	79.44±4.58	*	80.87±5.91	*
heart-c	83.97±0.63	83.17±0.76	*	84.05±0.6	
heart-h	83.84±0.63	82.58±0.89	*	83.9±0.62	
heart-statlog	90.16±5.79	82.32±9.03	*	90.85±5.12	
hepatitis	86.68±11.52	61.24±15.08	*	88.41±10.91	
hypothyroid	87.51±6.12	75.7±9	*	87.81±6.09	
ionosphere	94.54±4.66	88.45±5.98	*	93.4±4.79	
iris	98.93±1.91	90.8±2.66	*	98.64±2.17	
kr-vs-kp	99.82±0.15	99.82±0.24		95.16±1.2	*
labor	98±7.5	64.29±15.74	*	98.17±7.36	
letter	98.39±0.15	93.03±0.43	*	96.88±0.21	*
lymph	89.34±2.64	82.47±7.83	*	90±1.71	
mushroom	100±0	99.88±0	*	99.79±0.07	*
primary-tumor	78.76±1.64	72.97±1.4	*	78.87±1.76	
segment	99.24±0.28	97.16±0.72	*	98.5±0.41	*
sick	96.6±2.35	93.3±4.59	*	95.82±2.4	*
sonar	82.68±9.99	69.9±10.07	*	84.17±9.52	
soybean	99.77±0.34	83.83±1.89	*	99.73±0.34	
splice	99.4±0.34	96.81±1.09	*	99.45±0.28	
vehicle	85.5±2.79	82.25±3.55	*	80.31±3.09	*
vote	98.14±2.13	96.29±2.98	*	96.95±2.14	
vowel	97.67±0.77	88.43±2.02	*	95.58±1.12	*
waveform-5000	92.96±0.84	84.42±1.5	*	95.29±0.68	v
zoo	89.48±2.37	88.33±2.79	*	89.48±2.37	
Mean	89.85±3.45	82.41±4.83		89.50±3.52	

Table 5. Experimental results: two-tailed t-test with the 95% confidence level

	C4.5-NB	C4.5	NB
C4.5-NB	-	0	2
C4.5	35	-	28
NB	11	2	-

(third column, first row). Besides, the average value of C4.5-NB (89.85) is much higher than that of NB (89.50).

4 Conclusions and Future Work

Due to simplicity and high performance, C4.5 and NB are two very important algorithms for addressing the classification problems. In this paper, we present a combined classification algorithm based on C4.5 and NB, simply C4.5-NB. The experimental results show that the classification accuracy of C4.5-NB significantly outperforms C4.5 and NB.

In learning C4.5-NB, how to define the weights is crucial. In this paper, we use the accuracy of basic classification algorithms to estimate the weights directly from the training data. We believe that the use of more sophisticated methods could improve the performance of the current C4.5-NB and make its advantage stronger. This is the main research direction for our future work.

Acknowledgements

This paper is supported by the Research Foundation for Outstanding Young Teachers, China University of Geosciences (Wuhan) (No. CUGQNL0830).

References

1. Mitchell, T.M.: Decision tree Learning. Chapter 3 in Machine Learning. McGraw-Hill, New York (1997)
2. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
3. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1, 81–106 (1986)
4. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
5. Langley, P., Iba, W., Thomas, K.: An analysis of Bayesian classifiers. In: Proceedings of the Tenth National Conference of Artificial Intelligence, pp. 223–228. AAAI Press, Menlo Park (1992)
6. Friedman, G., Goldszmidt: Bayesian Network Classifiers. Machine Learning 29, 131–163 (1997)
7. Merz, C., Murphy, P., Aha, D.: UCI repository of machine learning databases. In: Dept of ICS, University of California, Irvine (1997), `http://www.ics.uci.edu/mlearn/MLRepository.html`
8. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005), `http://prdownloads.sourceforge.net/weka/datasets-UCI.jar`
9. Zhang, H., Jiang, L., Su, J.: Hidden Naive Bayes. In: Proceedings of the 20th National Conference on Artificial Intelligence, AAAI 2005, pp. 919–924. AAAI Press, Menlo Park (2005)

10. Liang, H., Zhang, H., Guo, Y.: Decision Trees for Probability Estimation: An Empirical Study. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2006, pp. 756–764. IEEE Computer Society Press, Los Alamitos (2006)
11. Nadeau, C., Bengio, Y.: Inference for the generalization error. Advances in Neural Information Processing Systems 12, 307–313 (1999)
12. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159 (1997)
13. Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning 45, 171–186 (2001)
14. Jiang, L., Zhang, H., Cai, Z., Su, J.: Learning tree augmented naive bayes for ranking. In: Zhou, L.-z., Ooi, B.-C., Meng, X. (eds.) DASFAA 2005. LNCS, vol. 3453, pp. 688–698. Springer, Heidelberg (2005)
15. Jiang, L., Zhang, H., Cai, Z.: Discriminatively Improving Naive Bayes by Evolutionary Feature Selection. Romanian Journal of Information Science and Technology 9(3), 163–174 (2006)

A Keyword Extraction Method Based on Lexical Chains

Xiao-yu Jiang[1,2]

[1] School of Computer Science, Beijing Institute of Technology, Beijing, 100081, China
[2] Business School, Beijing Institute of Clothing Technology, Beijing, 100029, China
sxyjiangxiaoyu@bict.edu.cn

Abstract. A lexical-chain-based keywords extraction method from Chinese texts is proposed in this paper, and an algorithm for constructing lexical chains based on HowNet knowledge database is given in the method, lexical chains are firstly constructing by calculating he semantic similarity between terms, then keywords are extracted according to the lexical chain's intensity, the terms' entropy and position. Unknown word recognition and semantic information between terms are considered in this method, which can significantly improve the effectiveness of keywords extraction. The preliminary experimental results show that the method gets better performance than other methods both in accurate-matching test and approximate-matching test.

Keywords: lexical chain, keyword extraction, word similarity.

1 Introduction

Keywords can be considered as brief summaries of a text. Therefore it is possible to think of them as a set of phrases semantically covering most of the text. Although a summary of a text is capable of providing more information about the text than keywords of the text, the summary may not be suitable for some applications due to the complex structure of sentences. Keywords are not replacements for summarization but alternative summary representations that could be consumed by other applications more easily. Since they are concise representations of the underlying text, it is possible to use them in different applications such as indexing in search engines or text categorization [1]. Keywords enable readers to decide whether a document is relevant for them or not. They can also be used as low cost measures of similarity between documents.

Keywords are most familiar in the context of journal articles, but many other types of documents could not benefit from the use of keywords, including web pages, email messages, news reports, magazine articles, and business papers. The vast majority of documents currently do not have keywords. Although the potential benefit is large, it would not be practical to manually assign keywords to them. This is the motivation for developing algorithms that can automatically supply keywords for a document.

2 Related Work

In text mining, keyword acquisition falls into two ways: keyword assignment and keyword extraction. They differentiate in the source of the keywords: keyword

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 360–367, 2008.
© Springer-Verlag Berlin Heidelberg 2008

assignment acquires keywords from predefined corpus or thesaurus, and keyword extraction from the source document.

As keyword assignment is domain dependence, and is not easy to transfer and expand. In recent years Domestic and foreign researchers keen on the research of the keyword extraction technology. The study of keyword extract starts earlier in abroad than that at home, and has established a number of experimental systems. Turney[2] applied the decision tree algorithm C4.5 and a genetic algorithm (GenEx)for keyword extraction, two features were used for these algorithms: word position in text and word frequency. Experimental result has shown that using bagging decision trees in C4.5 algorithm improves the performance of C4.5 for keyword extraction. ; The KEA[3] system uses *TFIDF* and the distance from the beginning of the document as features for learning which keywords to extract, and uses a Naïve Bayes classifier to assign weights to features.

As there has no explicit border between Chinese words, this increases a certain difficulty to the keyword extraction from Chinese document, YangWenfeng[4] presented an efficient algorithm to extract the max-duplicated strings by building PAT-tree for the document, so that the keywords can be picked out from the max-duplicated strings by their SIG values in the corpus, but this method is too complicated and needs a lot of space in the course of operation; LiShujian[5] proposed a keyword indexing method based on maximum entropy model, as the choice of characteristics and the estimated parameters is inaccurate, the maximum entropy model is not satisfactory in the application of keyword extraction.

3 Lexical Chain

A lexical chain is a sequence of related words in the document, spanning short (adjacent words or sentences) or long distances (entire document). A chain is independent of the grammatical structure of the document and in effect it is a list of words that captures a portion of the cohesive structure of the document. A lexical chain can provide a context for the resolution of an ambiguous term and enable identification of the concept that the term represents. *HowNet* is one lexical resource that may be used in the identification of lexical chains. Morris and Hirst [7] were the first researchers to suggest the use of lexical chains to determine the structure of texts. In recent years a number of researchers have used lexical chains in information retrieval, information extraction, automatic summarization and related areas.

3.1 Similarity Computation Based on HowNet

Liu Qun[6] defined word similarity as two words that can substitute for each other in the same context and still keep the sentence syntactically and semantically consistent. This is very close to our definition of synonyms. Thus, in this work, we will directly use the similarity function provided by Liu Qun, which is stated below.

A word in *HowNet* is defined as a set of concepts, and each concept is represented by primitives. We describe *HowNet* as a collection of n words, W:

$$W = \{w_1, w_2, \ldots, w_n\}.$$

Each word w_i is, in turn, described by a set of concepts S

$$w_i = \{S_{i1}, S_{i2}, \ldots, S_{ix}\},$$

Each concept S_i is, in turn, described by a set of primitives: $S_i = \{p_{i1}, p_{i2}, \ldots, p_{iy}\}$.

For each word pair, w_1 and w_2, the similarity function is defined by

$$Sim(w_1, w_2) = \max_{i=1..n, j=1...m} Sim(S_{1i}, S_{1j}) \tag{1}$$

where S_{1i} is the list of concepts associated with w_1 and S_{2j} is the list of concepts associated with w_2.

As any concept, S_i is represented by its primitives. The similarity of primitives for any p_1 and p_2 of the same type can be expressed by the following formula:

$$Sim(p_1, p_2) = \frac{\alpha}{Dis(p_1, p_2) + \alpha} \tag{2}$$

where α is an adjustable parameter with a value of 1.6 according to Liu[6], $Dis(p_1, p_2)$ is the path length between p_1 and p_2 based on the semantic tree structure.

Based on the DEF descriptions in *HowNet*, different primitive types play different roles, and only some are directly related to semantics. To make use of both semantic and syntactic information, the similarity between two concepts should take into consideration all the primitive types with weighted considerations; and thus, the formula is

$$Sim(S_1, S_2) = \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} Sim_j(p_{1j}, p_{2j}) \tag{3}$$

where β_i ($1 \le i \le 4$) is a weighting factor, where the sum of $\beta_1+\beta_2+\beta_3+\beta_4$ is 1 and $\beta_1 \ge \beta_2 \ge \beta_3 \ge \beta_4$. The distribution of the weighting factors is given for each concept a priori in *HowNet* to indicate the importance of primitive *pi* for the corresponding concept *S*. The similarity model given here is the basis for building a synonym set where β_1 and β_2 represent the semantic information, and β_3 and β_4 represent the syntactic relation.

3.2 Building Lexical Chain

The calculation model and algorithm of lexical chain based on *WordNet* have been proposed by Morris [7] according to the relationship between words. In *WordNet*, English words are organized into synonym sets in accordance with inherent semantic concept which they represent. In this paper, we adopt the *HowNet* to determine the relationship between the Chinese words, and build lexical chain. During the calculation, only the nouns which appear in the *WordNet* are selected to be candidate words for keyword extraction, in order to improve the precision of keyword extraction, we choose all the nouns, some verbs and adjectives (frequency greater than expected

threshold) which appear in the *HowNet*, and unknown new terms to be candidate words, the algorithm of the lexical chains' construction is presented as follows:

① Word segmentation, part-of-speech tagging and unknown word recognition for document set, and calculate the term frequency(*TF*) and document frequency(*DF*) for each word ;

② Because some field terms are not include in HowNet, and these terms are relatively important, so these unknown words whose *TF* is greater than threshold δ (δ is set to 3)will be as a separate lexical chain L_0;

③ Select all the nouns, some verbs whose *TF* is greater than threshold δ as candidate set of words W^*, $W^*=\{w_1, w_2,\ldots, w_n\}$, and an initial lexical chain L_1 is constructed using w_1 ;

④ choose word w_i ($i\in[2,n]$) in order from W^*, and calculate the semantic similarity $S(w_i,L_j)$ between w_i and each lexical chain (In addition to lexical chain L_0)according to the formula (4), that is the average of similarity between w_i and all the words of the lexical chain;

$$S\left(w_i, L_j\right)=\frac{1}{N}\sum_{k=1}^{N} Sim\left(w_i, w_k\right)=\frac{1}{N}\sum_{k=1}^{N}\max_{i=1..n,\, j=1...m} Sim\left(S_{1i}, S_{2j}\right) \tag{4}$$

where N is the number of words in the L_j, w_k is the word of L_j, $1\le k\le N$;

⑤ If the greatest similarity $S(w_i,L_k)$ is greater than the similarity threshold, w_i will be insert into lexical chain L_k;

⑥ If the greatest similarity $S(w_i, L_k)$ is less than the similarity threshold, a new lexical chain will be created and w_i will be insert into this lexical chain;

⑦ Repeat the process③~⑥, until all the candidates are calculated.

In above algorithm, it is not difficult to see that the choice of the similarity threshold ζ is proportional to the number of the lexical chains, that is, the bigger ζ is, the more the number of lexical chains.

3.3 Weight Calculating of Lexical Chain

During calculating the importance of lexical chain, the following factors [7] are considered: ①the length of lexical chain (the number of words in the lexical chain); ②the initial weight of each word in the lexical chain; □the scope of document which lexical chain covers, the bigger scope, the more content included in lexical chain; □the distribution density of words in the lexical chain, the more concentrated the distribution, the higher of the importance.

So far, the construction of lexical chains is completed, and the corresponding weight is assigned to each lexical chain. Each text is express as $T=\{T_1, T_2, \ldots, T_n\}$,where T_i is the weight of lexical chain L_i ($0\le i\le n$). The greater of the weight of lexical chain, the stronger of the ability to express the theme of the document, on the contrary, the smaller of the weight, the more distant from the theme. A threshold was predefined, and several

strongest lexical chains (certainly include the lexical chain which unknown words belong to) are selected, and keywords should be extracted from these lexical chains.

4 Keyword Extraction

In fact, the lexical chain L_i constructed according to above algorithm is a set with a number of similar semantic terms, which of these terms will be choose as keywords need to consider the following four attributes:

① Position of term w_i first appears: the ratio of the number of words which appear before the first W_i and the total number of words in the document, the value of this attribute in the 0-1;
② Area at which w_i locate: this property is based on the following assumptions: terms appear in the titles, summary and chapter headings of document have larger weight and have larger possibility to be keywords than other terms;
③ Strength of lexical chain: that is the weight of lexical chain which w_i belong to, the greater of the weight, the stronger of the ability to express the theme of the document; on the contrary, the smaller of the weight, the weaker of the ability to express the theme of the document;
④ Information Entropy: that is how much content of the document w_i contained ; if w_i appears in all the sentences or documents, the information entropy will be very little; if w_i appears in individual sentences or documents, the information entropy will be very little, the information entropy will be great, such thinking is consistent with *TFIDF*, the calculation method such as the formula (5):

$$E_i = \frac{1}{log(M)} \sum_{j=1}^{M} \left[\frac{f_{ij}}{df_i} log\left(\frac{f_{ij}}{df_i} \right) \right] \tag{5}$$

where E_i is the information entropy of w_i; M is the total number of sentences in the single document or the number of documents in document set; f_{ij} is the frequency of w_i in the sentence S_j or document D_j ; df_i is the number of sentences or documents in which w_i appears.

The position of term first appears, the area of term locate, the strength of lexical chain which term belong to and information entropy four attributes are considered complete, and a calculation method of term's weight is proposed as formula (6):

$$Weight_i = \alpha * log(f_i + 1.0) * (1 + E_i) + \beta * T_i + \gamma * \frac{Length_i}{Length} + \eta * Area_i \tag{6}$$

where $Weight_i$ is the weight of w_i; f_i is the term frequency of w_i; T_i is the weight of lexical chain which w_i belong to; $Length_i$ is the number of words which appear before the first w_i in the document; *Length* is the total number of words in the document; $Area_i$ is the weight of area where w_i locate at, if w_i appears in the title of document, $Area_i$=5; if w_i appears in the summary, $Area_i$=4;if w_i appears in the summary, $Area_i$=2; otherwise, $Area_i$=0.5;α,β,γ and η are four regulators which are used to balance the four properties during the calculation of weight, under normal circumstances these four regulators are all set to 1.

When the weight of each word in the lexical chain T_i is calculated, T_i can be expressed as $T_i = \{t_{i1}, t_{i2} \ldots t_{im}\}$, and t_{ij} is the weight of w_j which belong to L_i. After words of all lexical chains constructed in section 3.3 are calculated, all words will be sorted in descending order in accordance with its weight, and these words with greatest weight are extracted as keywords until the number of keywords is greater than the predefined number.

5 Experiments and Discussion

5.1 Experimental Data

Keywords are prepared by authors in all academic papers, so we download 100 papers about "automatic summarization" as experimental data from China Journal Net (http://www.cnki.net). As we only extract those terms as keywords which must appear in the thesis, and moreover, some keywords prepared by authors don't be necessarily arise in the academic papers, so each of these keywords should be replaced with semantic similar term if there exist similar term with it in the paper; if there does not exist, these keywords would be deleted from keywords collection. In the experiment, we found that more than 94% keywords similar to this situation can find its semantic similar term directly from the article.

5.2 Evaluation Metric

It is necessary to evaluate the performance of keyword extraction when keywords are extracted from document automatically. The most commonly used method of evaluation is to match the keywords extracted automatically and the keywords selected by human, two different metrics (precision and recall) are used to compare the effect of extraction.

$$Precision = \frac{|A \cap B|}{|A|} \tag{7}$$

$$Recall = \frac{|A \cap B|}{|B|} \tag{8}$$

where $|A|$ is the number of keywords extracted automatically; $|B|$ is the number of keywords selected by author; $|A \cap B|$ is the size of the Intersection between A and B.

5.3 Experimental Methods and Results

In order to verify the feasibility of the algorithm proposed in this paper, the comparative experiment is implemented between the keywords extraction algorithm based on lexical chains (*Method Lexical Chain*) and the keywords extraction algorithm based on frequency and area of terms (*Method Frequency and Area*).The number of keyword extracted automatically has an important influence on the precision and recall, so based on accurate-matching ,the above two algorithms are tested in different number of keywords, the comparative test result s are shown in Figure 1 and Figure 2.

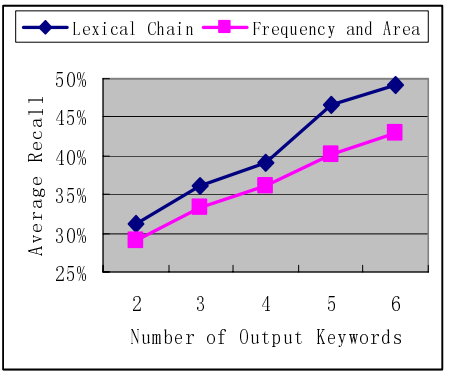


Fig. 1. Comparative testing result of precision based on accurate-matching

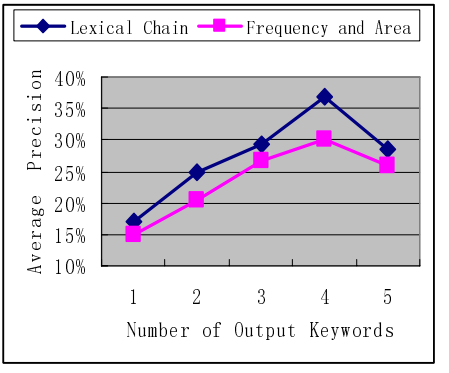


Fig. 2. Comparative testing result of recall based on accurate-matching

5.4 Experimental Analysis

From Figure 1 and Figure 2, we can find that method lexical chain gets better performance than method frequency and area, but it has low precision and recall which are less than 50%.In the experiment, we find accurate-matching has a certain drawbacks, for example, we believe that one keyword “Word Segmentation” prepared by the author of the article does not match to the keyword “Chinese Word Segmentation” extracted automatically, this is obviously unreasonable, so we adopt a approximate-matching approach, that is, we think it is suited if there exists inclusion relationships or semantic similarity is greater than predefined threshold between keyword prepared by author and keyword extracted automatically. The comparative test results are shown in Figure 3 and Figure 4.

From Figure 3 and Figure 4, we can find that approximate-matching gets better effect than accurate-matching, also it shows a big advantages of method lexical chain.

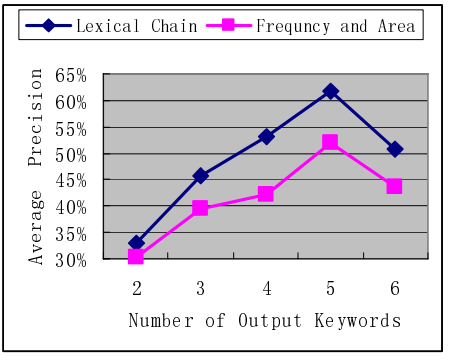


Fig. 3. Comparative testing result of precision based on approximate-matching

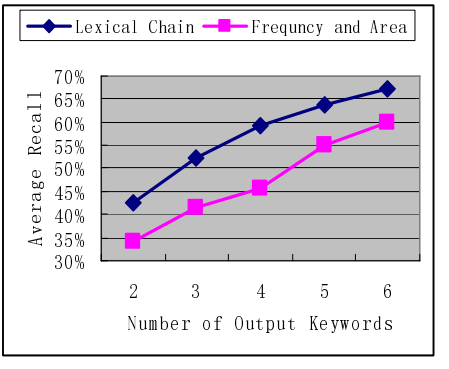


Fig. 4. Comparative testing result of recall based on approximate-matching

Analysis shows that main reasons lie in: ①method frequency and area has more reliance on the identification of titles and headings than method lexical chain; ②these keywords predefined by author do not always appear in the title or summary of article. The weight of terms in article will be calculated in method lexical chain, and the chapter headings and titles of article has only a small part of weight, the low accuracy of identify the headings and titles has limited impact on the performance, so method lexical chain is more reasonable and effective than method frequency and area, it is also consistent with the experimental results.

6 Conclusions

As Lexical chain is composed by a series of words with semantic relevance, terms in the document are first organized into lexical chains on the basis of unknown word identification; the strength of lexical chain, information entropy, the position and other attributes are combined to improve the performance of keyword extraction. The method is domain independence and can be used to extract keywords from single document or document set. Experimental results show that the method gets better performance both in accurate-matching test and approximate-matching test. Next we will study how to apply this keywords extraction algorithm to the generation of summary automatically.

References

1. Xu, W., Wen, Y.: A Chinese Keyword Extraction Algorithm Based on TFIDF. Information System 31(2), 298–302 (2008)
2. Turney, P.D.: Learning to Extract Keyphrases from text. National Research Council, Canada, NRC Technical Report ERB-1057 (1999)
3. Witten, I.H., Paynter, G.W., Frank, E.: KEA: Practical Automatic Keyphrase Extraction. In: Proceedings of the 4th ACM Conference on Digital Libraries, Berkeley, California, US, pp. 254–256 (1999)
4. Yang, W.: Chinese Keyword Extraction based on Max-duplicated Strings of the Documents. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, pp. 439–440 (2002)
5. Li, S., Wang, H., Yu, S.: Research on Maximum Entropy Model for Keyword Indexing. Chinese Journal of Computers 27(9), 1192–1197 (2004)
6. Liu, Q., Li, S.: Word Semantic Similarity Computation based on Hownet. In: Proceedings of the 3rd Chinese Lexical Semantics Workshop. Academia Sinica, Taipei (2002)
7. You, W., Li, S., Li, T.: A Text Filtering Module Based on Lexical Chain. Application Research of Computers 20(9), 32–35 (2003)
8. Morris, J., Hirst, G.: Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. Computational Linguistics 17(1), 21–48 (1991)

A Method of the Extraction of Texture Feature

Haifang Li, Lihuan Men, and Junjie Chen

College of Computer and Software, Taiyuan University of Technology, Taiyuan, Shanxi, 030024, China
mlh115101@126.com, sxlhf123@163.com

Abstract. In order to understand the emotional information of the color image, research focus has been shifted from designing sophisticated low-level feature extraction algorithms to reducing the 'semantic gap' between the visual features and the richness of human perception. In this paper, we firstly get the ROI using the Eye tracker and divide every image into two regions including Regions of Interest (ROI) and Non- Regions of Interest (Non-ROI). Secondly, we use the analytical hierarchy process (AHP) to provide a systematical way to evaluate the fit weights of ROI and Non-ROI. Finally, using the improved GLCM, we extract the texture feature of the two regions including ROI and Non-ROI, and get the whole texture feature. The algorithm is tested that the average detection rate of the proposed method is up to the same method using GLCM.

Keywords: Regions of Interest (ROI), Texture feature, Gray Level Co-occurrence Matrix (GLCM), Analytical Hierarchy Process (AHP), Weight.

1 Introduction

1.1 Background

There are different emotional information in different images, and in order to understand the emotional information of the color image, research focus has been shifted from designing sophisticated low-level feature extraction algorithms to reducing the 'semantic gap' between the visual features and the richness of human perception.

The relationship between feature and emotion is closely tied to feature preferences. In particular, feature preferences are associated with whether features elicit positive or negative feelings. While particular features have been found to be highly preferred regardless of age, racial group, or culture (Adams & Osgood, 1973, Eysenck, 1941), there is some evidence that feature preferences may be culturally-based. So feature extraction is an indispensable step in the processing, which lays the basis for both retrieval to the database format and the subsequent cluster stage. The robustness and accuracy of the feature extraction directly influences the success of the final results.

1.2 Regions of Interest (ROI) and Eye Tracking System

Regions of Interest (ROI) has been among one of the most important research topics for the past several years and it still remains to be an open problem. ROI can attract users'

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 368–377, 2008.
© Springer-Verlag Berlin Heidelberg 2008

attention and can express the emotion of the whole image effetely [12], [13], [14]. So we assume in this paper that focused region in the scene always corresponds to the ROI.

To get accurate emotion of images, in this paper we will use Eye tracking system, an apparatus which records of both eyes at a selectable frame rate from 5 to 30 Hz and stored the records to computer disk for offline analysis with commercial eye tracking software [1]. Eye movements were tracked about 3 axes, and orientations were described as Fick angles. These were derived from the projection of basis vectors of an eyeball fixed coordinate system onto a head-fixed one [2]. Pupil size and position were calculated using a robust algorithm for image feature detection based on the Hough Transform modified for ellipses [3], [4], [5]. Use of ellipse fitting has been shown to better approximate the pupils of most eyes and compensates for geometric distortion from image projection [6]. The results of that study [1] illustrate that the Eye tracking viewing device enables objective data collection for the study of eye movements and ocular motor control.

1.3 The Main Work of This Paper

In this study, we presented a new method for getting the ROI using Tobii Eye tracking system which provides a means of interaction with users and shows the eye movement trajectory. We define the regions eyes stay longer time as the ROI. And we have gotten ROIs of 852 images that were disposed with 100 people by Tobii Eye tracking system. As Fig. 2 (a) And the ROIs can represent the main emotion of the images effectively.

And then, we provide a systematical way (AHP) to evaluate the fit weights of ROI and Non-ROI. Finally, we extract the texture feature of the two regions including ROI and Non-ROI separately and to get the whole texture feature. Finally, the algorithm is tested that the average detection rate of the proposed method is up to the same method using GLCM.

2 Weight Distribution Using Analytical Hierarchy Process (AHP)

2.1 A Brief Review of AHP

The process of AHP includes three stages of problem solving: decomposition, comparative judgments, and synthesis of priority. The decomposition stage aims at the construction of a hierarchical network to represent a decision problem, with the top level representing overall objectives and the lower levels representing criteria, sub-criteria, and alternatives. With comparative judgments, users are requested to set up a comparison matrix at each hierarchy by comparing pairs of criteria or sub-criteria. A scale of values ranging from 1 (indifference) to 9 (extreme preference) is used to express the users preference. Finally, in the synthesis of priority stage, each comparison matrix is then solved by an eigenvector method for determining the criteria importance and alternative performance. The following list provides a brief summary of all steps involved in AHP applications:

(1). Specify a concept hierarchy of interrelated decision criteria to form the decision hierarchy.
(2). For each hierarchy, collect input data by performing a pairwise comparison of the

decision criteria.
(3). Estimate the relative weightings of decision criteria by using an eigenvector method.
(4). Aggregate the relative weights up the hierarchy to obtain a composite weight which represents the relative importance of each alternative according to the decision-maker's assessment.

One major advantage of AHP is that it is applicable to the problem of group decision-making. In group decision setting, each participant is required to set up the preference of each alternative by following the AHP method and all the views of the participants are used to obtain an average weighting of each alternative. [7]

2.2 Weight Distribution Using AHP

We view an image as a compound object containing multiple component objects which are then described by several semantic descriptions according to a three-level concept hierarchy. The concept hierarchy, shown in Fig. 1, is used to assign the semantics to a ROI for an input image. According to the hierarchy, the method for involving the image of ROI to an image database by AHP is proposed in this study. For the sake of illustration convenience, the classification hierarchy is abbreviated as WRC hierarchy.

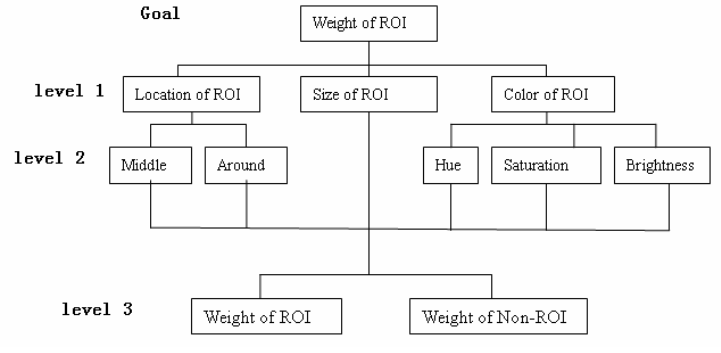


Fig. 1. The concept hierarchy of the weight of ROI for interpreting an input image

There are three subjects in the top level of WRC hierarchy. Each top-level subject corresponding to the properties of ROI is then divided into several sub-subjects corresponding to the properties of them, and each sub-subject is again decomposed into several Level 3 subjects corresponding to the ROI or Non-ROI. A path from the root to each leaf node forms a semantic description, and multiple semantic descriptions are possible to interpret a ROI according to different aspects of user notion. A question arises naturally: is the weight of each path code of an image object equivalent? The answer to the problem is of course no. Some semantic descriptions are obviously more important than others for a specific image object. For example, the semantic description "size" is more important than the code with the semantic description "color" for the image object in Fig. 2 (b) according to the authors' opinion.

2.3 Weight of ROI Representation

Assume the path codes of the properties of ROI (low-level) hierarchy are numbered from 1 to n. Given an image I, the content of the image is represented by a semantic vector which is defined as

$$I=(s1,s2,\ldots\ldots,sn),\ \sum_{i=1}^{n} s_i=1 \tag{1}$$

where si denotes the weighting of the ith path code. Although the value of n is large, in any vector representing an image, the vast majority of the components will be zero. The reason is that the number of objects perceived in an image is generally small.

The judgment of the importance of one semantic description over another can be made subjectively and converted into a numerical value using a scale of 1–9 where 1 denotes equal importance and 9 denotes the highest degree of favoritism. Table 1 lists the possible judgments and their representative numerical values. The numerical values representing the judgments of the pairwise comparisons are arranged to form a reciprocal matrix for further calculations. Users are required to adopt a top-down approach in their pairwise comparisons. Given an image, the first step of the classification process using AHP is to choose the large classification codes and evaluate their relative importance by performing pairwise comparisons. For example, Fig. 2 (a) containing a ROI image is the target of weight distribution. In this case, the image can be classified into three Level 1 property categories—"Location" L, "Size" S, and "Color" C, Fig. 2 (b) is the corresponding Level 1 reciprocal matrix M1 for judging the relative importance of the three property descriptions. The entries of M1 can be denoted as

$$M_1=\begin{array}{c} \\ L \\ S \\ C \end{array}\begin{array}{c} \begin{array}{ccc} L & S & C \end{array} \\ \begin{bmatrix} w_L/w_L & w_L/w_S & w_L/w_C \\ w_S/w_L & w_S/w_S & w_S/w_C \\ w_C/w_C & w_C/w_S & w_C/w_C \end{bmatrix} \end{array} \tag{2}$$

where wL, wS, and wC are the relative importance values (defined in Table. 1) for the three semantic descriptions L, S, and, C, respectively. Level 1 weightings of the three semantic descriptions are then obtained from M1.

Without lose of generality, let l, m, and n be the number of Level 1 semantic descriptions, the number of Level 2 semantic descriptions for each Level 1 description, the number of Level 3 semantic description for each Level 2 description, respectively. For each row of Level 1 reciprocal matrix M1, we can define a weighting measurement as

$$r_i^1=\sqrt[l]{\frac{a_i^1}{a_1^1}\times\frac{a_i^1}{a_2^1}\times\cdots\times\frac{a_i^1}{a_l^1}},i=1,\ldots,l, \tag{3}$$

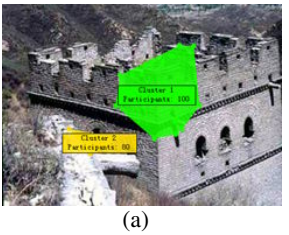
where a_i^1 is the relative important value of the ith Level 1 semantic description. Then Level 1 weightings are determined by

$$w_i^1 = r_i^1 \Big/ \sum_{j=1}^{l} r_j^1, i=1,...,l. \tag{4}$$

Similarly, we can compute Level 2 weightings $w_{i,j}^2$, j=1,……,m for the ith Level 1 semantic description and Level 3 weightings $w^3_{i,j,k}$, k=1,…,n for the ith Level 1 semantic description and the jth Level 2 semantic description. Finally, the entry p of the semantic vector defined in Eq. (1) is computed as

$$w_p = (i-1)\times l + (j-1)\times m + k = w_i^1 \times w^2_{i,j} \times w^3_{i,j,k}. \tag{5}$$

Note that the number of reciprocal matrixes for the image is $l \times$m $\times$ n and it is actually equal to the number of path codes used to describe the face image. It would be too cumbersome to classify an image using AHP if the value of $l \times$ m $\times$ n is very large.



(a)

	Location	Size	Color
Location	1	4	1/2
Size	1/4	1	1/3
Color	2	3	1

(b)

Fig. 2. (a) the image with ROI; (b) the corresponding reciprocal matrix with respect to (a) for calculating the local weightings of Level 1 semantic descriptions in interpreting the weights of the image

To get accurate result, we make a survey to get all of the matrixes. And our research covered 100 people including 54 females and 46 males. The finally result is that weight of ROI is 0.6459 and weight of Non-ROI is 0.3541.

Table 1. Pairwise comparison judgments between semantic descriptions A and B

Judgment	Values
A is equally preferred to B	1
A is equally to moderately preferred over B	2
A is moderately preferred over B	3
A is moderately to strongly preferred over B	4
A is strongly preferred to B	5
A is strongly to very strongly preferred over B	6
A is very strongly preferred over B	7
A is very strongly to extremely preferred over B	8
A is extremely preferred to B	9

3 Gray Level Co-Occurrence Matrix and Improved Gray Level Co-Occurrence Matrix

3.1 Texture Analysis Using Gray Level Co-Occurrence Matrix（GLCM）

In this section, the GLCM is presented as representative of the statistical approaches to texture analysis. The GLCM of an image is an estimate of the second-order joint probability, P_δ (i, j) of the intensity values of two pixels (i and j), a distance δ apart along a given direction θ, i.e., the probability that i and j have the same intensity. This joint probability takes the form of a square array P_δ, with row and column dimensions equal to the number of discrete gray levels (intensities) in the image being examined. If an intensity image were entirely flat (i.e. contained no texture), the resulting GLCM would be completely diagonal. As the image texture increases (i.e. as the local pixel intensity variations increase), the off-diagonal values in the GLCM become larger.

The pixel intensity resolution of the steel surface grayscale images used in this paper is 8-bit, which result in GLCMs with dimensions of 256 rows × 256 columns for a given displacement vector. Finding GLCMs for all distances (δ) and angles (θ) would require a prohibitive amount of computation. Haralick et al. [8] suggested using GLCMs calculated from four displacement vectors with δ =1, or 2 pixels, and θ =0°, 45°, 90°, and 135°. In this example, only one GLCM was calculated for each of the 35 grayscale steel surface images using a single displacement vector with δ = 1, and θ = 135° [(xlag,ylag)=(1,1)].

Haralick et al. [8] proposed a quantitative analysis of the GLCM through 14 textural descriptors calculated from P_δ, although typically only a few of these are widely used [9], [10], [11]. In this paper, four of the most commonly used descriptors (the angular second moment, contrast, correlation, and entropy) are used to extract textural features from the 35 GLCMs of the steel surface grayscale image data set.

$$\text{Angular Second Moment} = \sum_{i=1}^{n}\sum_{j=1}^{n}\{P_\delta(i,j)\}^2 \tag{6}$$

$$\text{Contrast} = \sum_{k=0}^{n-1} k^2 \sum_{i=1}^{n}\sum_{\substack{j=1 \\ |i-j|=k}}^{n} P_\delta(i,j) \tag{7}$$

$$\text{Correlation} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} i\cdot j P_\delta(i,j) - \mu_x\mu_y}{\sigma_x\sigma_y} \tag{8}$$

$$\text{Entropy} = -\sum_{i=1}^{n}\sum_{j=1}^{n} P_\delta(i,j)\log\{P_\delta(i,j)\} \tag{9}$$

Where the means and variances in the x and y direction are given by

$$\mu_x = \sum_{i=1}^{n} i \sum_{j=1}^{n} P_\delta(i,j), \qquad \mu_y = \sum_{j=1}^{n} j \sum_{i=1}^{n} P_\delta(i,j),$$
$$\sigma_x = \sum_{i=1}^{n} (i-\mu_x)^2 \sum_{j=1}^{n} P_\delta(i,j), \qquad \delta_y = \sum_{j=1}^{n} (j-\mu_y)^2 \sum_{i=1}^{n} P_\delta(i,j) \tag{10}$$

3.2 Improved Gray Level Co-Occurrence Matrix

In this section, we extract features from ROI and Non-ROI separately using GLCM, and secondly, we evaluate the fit weights of ROI and Non-ROI using AHP. Finally, we fusion the features of ROI and Non-ROI to get the whole texture feature.

For example: we extract texture features of ROI, and

Angular Second Moment-ROI=A1,Contrast-ROI =B1,Correlation-ROI =C1, Entropy-ROI =D1,Angular Second Moment-Non-ROI=A2,Contrast-Non -ROI =B2, Correlation-Non -ROI =C2, Entropy-Non -ROI =D2.

The last result is that

Angular Second Moment =A1×0.6459+A2×0.3541,Contrast = B1×0.6459+ B2×0.3541, Correlation = C1×0.6459+ C2×0.3541, Entropy = D1×0.6459+ D2×0.3541.

4 Experimental Results

4.1 Datasets

The effectiveness of the proposed algorithm is evaluated through two datasets, one is the 852 images from CAPS image storehouse of Chinese Academy of Science and another is the predisposed dataset that the images from CAPS image storehouse are disposed by Eye tracking system. The dataset consists of 852 images, which are categorized into several emotions, such as joy, sadness, fear, disgust etc. Each of the categories contains about 100 images of essentially the same semantic concept, which serves as the groundtruth.

4.2 Experiments and Results

In this section we provide two experiments to illustrate the advantages of the proposed new algorithm. We use Self-Organizing Map (SOM) to verify the effectiveness of the approach presented in this article. The SOM algorithm is only one way to cluster the data, and it has many variants. There are a lot of completely different clustering algorithms and we select one of it to help us fulfill the experiments. We carry on two experiments as follows. The computer system is composed of 1.7 GHz processor and 256 M memory, Windows XP operating system, Matlab 7.0 programming language.

4.2.1 Using the Texture Feature by GLCM

Firstly, we have to extract the texture feature using GLCM and make all of the features into a data file (texture1label.data), and the data file will be the Data Source to the SOM. Then, we can cluster the images into several categories of emotions by SOM. And Fig. 3 shows the U-Matrix and the Emotion labels using texture feature without weight.

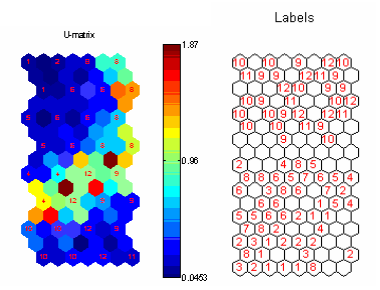


Fig. 3. U-Matrix and labels without weight

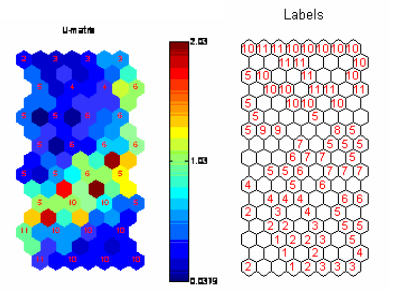


Fig. 4. U-Matrix and labels with weight

4.2.2 Using the Texture Feature by Improved GLCM

The process of the experiment is similar as the first one. The only difference is the data file. The data file (texture2label.data) here is made of the texture feature using the improved GLCM. Fig. 4 shows the U-Matrix and the Emotion labels using the texture feature with weights.

Unified distance matrix (U-matrix) visualizes distances between neighboring map units and helps to see the cluster structure of the map: high values of the U-matrix indicate a cluster border; uniform areas of low values indicate clusters themselves. The junior color (yellow or red or white) in the figures is the clearer part of the clustering.

Table 2. Number and Emotion

emotional number	emotional type	emotional number	emotional type
1	guilt	7	attentiveness
2	joviality	8	shyness,
3	hostility	9	fatigue,
4	sadness	10	serenity
5	fear	11	surprise
6	self-assurance	12	do not know

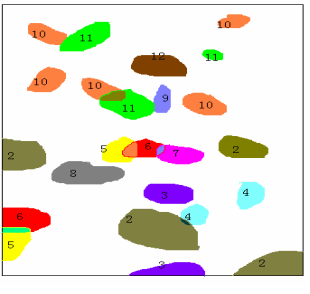


Fig. 5. Emotion Distribution Graph of Fig. 3

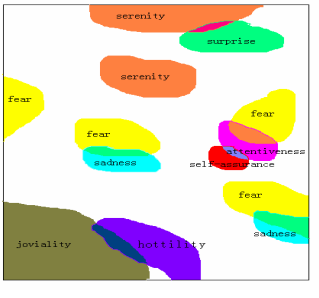


Fig. 6. Emotion Distribution Graph of Fig. 4

And labeling is used to categorize the units (or some units) by giving them names. The hexagons represent grid, and the numbers in the grids are on behalf of emotional types. The relationship of numbers and emotional types are shown in table.2. Fig. 5 and Fig. 6 are the Emotion Distribution Graph of Fig. 3 and fig. 4. Observation of them indicates that clustering of emotions in Fig.8 is more obvious.

According to the two Experiments and results, the algorithm is tested that the average effectiveness of the proposed method is up to the same method using GLCM.

5 Summary and Conclusion

In this paper, we have first presented the Eye tracker, the Definition of ROI, and weight distribution using AHP. And then we design two experiments that one is using GLCM to dispose the whole image and the other experiment is using the improved GLCM to disposed images which have been divided into ROI and Non-ROI.

We have shown that the new algorithm is very effective for improving the overall feature Extraction and emotion categories. With respect to efficiency, Eye tracker can show the ROIs in the initial processing, so that the succeeding processing stages are focusing on more detailed properties. With respect to accuracy, the ROIs are tested by 100 people, so that the final results are of sufficiently high accuracy. In this way, both accuracy and efficiency are simultaneously improved and combined into the same system design.

Acknowledgements. The work has been supported in part by the Key Project of The Natural Science Foundation of China under Grant (No. 60773004), and supported in part by the Key Project of The Natural Science Foundation of Shanxi Province (No.2006011030 and No.2007011050).

References

1. Ramey, N.A., Ying, H.S., Irsch, K.: A novel haploscopic viewing apparatus with a three-axis eye tracker. Journal of AAPOS.2008.01.019
2. Clarke, A.H., Ditterich, J., Druen, K., Schonfeld, U., Steineke, C.: Using high frame rate CMOS sensors for three-dimensional eye tracking. Behav. Res. Methods Instrum. Comput. 34, 549–560 (2002)
3. Hough, P.V.C.: Methods and means for recognizing complex patterns. U.S. Patent 3069654 (1962)
4. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM 15, 11–15 (1972)
5. Tsuji, S., Matsumoto, F.: Detection of ellipses by a modified Hough transform. IEEE Trans. Comput. 27, 777–781 (1978)
6. Moore, S.T., Haslwanter, T., Curthoys, I.S., Smith, S.T.: A geometric basis for measurement of three-dimensional eye position using image processing. Vision Res. 36, 445–459 (1996)
7. Cheng, S.-C., Chen, M.-Y., Chang, H.-Y., Chou, T.-C.: Semantic-based facial expression recognition using analytical hierarchy process (2007)

8. Haralick, R.M., Shanmugam, K., Dinstein, I.: IEEE Trans. Syst. Man Cybern. 3, 610–621 (1973)
9. Tomita, F., Tsuji, S.: Computer Analysis of Visual Textures. Kluwer Academic Publishing, Massachusetts (1990)
10. Weszka, J.S., Dyer, C.R., Rosenfeld, A.: IEEE Trans. Syst. Man Cybern. 5, 269–285 (1976)
11. Al-Janobi, A.: Pattern Recogn. 34, 171–180 (2001)
12. Schmid, C., Mohr, R.: IEEE Transactions on Pattern Analysis and Machine Intelligence. 19(5), 530 (1997)
13. Tian, Q., Sebe, N., Lew, M.S., et al.: Journal of Electronic Imaging. 10(4), 835 (2001)
14. Moghaddam, B., Biermann, H., Margaritis, D.: Defining image content with multiple regions of interest. In: Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries (CVPR 1999), p. 350 (1999)

A Transductive Learning Method for Interactive Image Segmentation

Jiazhen Xu, Xinmeng Chen, Yang Wei, and Xuejuan Huang

Computer School, Wuhan University,
Wuhan, China
jzxuwhu@yahoo.com.cn, xinmeng@whu.edu.cn,
william_yang2005@163.com, xjhuang@yahoo.cn

Abstract. The problem of interactive image segmentation is of great practical importance in image processing. Recently, a transductive framework has been developed which solves the problem from the perspective of manifold learning and shows promising results. In this paper, we extend this approach in two aspects. First, considering that the common interactive tools for user are broad brushes or region selection tools, it is hard to mark all the seeds accurately by hand. Our method is robust against noise, which releases the requirement of user and tolerant to a certain amount of user input faults. Secondly, we combine our method with prior statistical information implicitly provided by user input seeds, result in higher accuracy. Experiments results demonstrate improvements in performance over the former methods.

1 Introduction

Image segmentation has wide applications in computer vision, computer graphics and medical imaging. Fully automatic image segmentation has many intrinsic difficulties and is still a very hard problem. In many applications, such as image editing in computer graphics and organ segmentation in medical imaging, interactive approaches, where human operators provide seeds for the computational algorithms to perform segmentation, can not only overcome the inherent technical difficulties of fully automatic image segmentation, but may also be desirable because the operators in many of these applications may want to be able to control the segmentation process and results.

The general purpose of interactive segmentation can be reduced to the task that divides an image into two segments: *foreground* and *background* with user supplied seeds (Fig. 1). In most cases, extending binary segmentation method to multiple one is relatively straightforward.

1.1 Related Work

The graph cuts technique [3], addresses the foreground/background interactive segmentation in still images via min-cut energy minimization. The energy balances between the probability of pixels belonging to the foreground (likelihood) and the edge contrast, imposing regularization. The user-provided scribbles collect statistical information on pixels and also serve as hard constraints.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 378–385, 2008.
© Springer-Verlag Berlin Heidelberg 2008

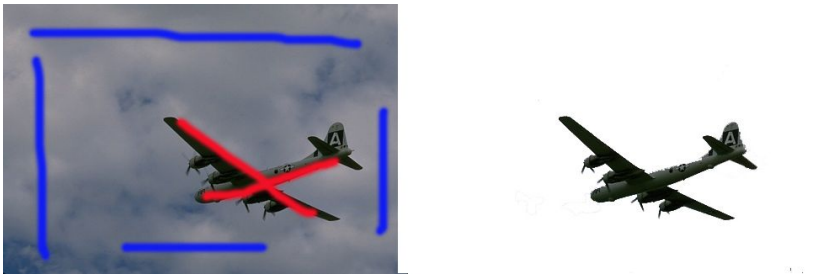


Fig. 1. Left: an input image with user-supplied strokes. Right: the foreground of segmentation.

The Grabcut algorithm [6], further simplifies the user interaction. Scribbles can be interactively added to improve the initial segmentation. Full color statistics are used, modeled as mixtures of Gaussians (here, in contrast, we use fast kernel density estimation), and these are updated as the segmentation progresses. This can help but also hurt by propagating segmentation errors.

The random walk approach proposed in [5] is inspired discrete potential theory and electrical circuits. It determines the probability that a random walker starting at each unlabeled pixel will first reach one of the pre-labeled pixels. By assigning each pixel to the label for which the greatest probability is calculated, a high-quality image segmentation may be obtained.

[4] reveals the intrinsic links with all the approaches above and brings up a tranductive framework covering these approaches (different in parameters). It views this task as a statistical transductive inference in which some pixels are already associated with given zones and the remaining ones need to be classified. This method relies on the Laplacian graph regularizer, a powerful manifold learning tool that is based on the estimation of variants of the Laplace-Beltrami operator and is tightly related to diffusion processes. Segmentation is modeled as the task of finding matting coefficients for unclassified pixels given known matting coefficients for seed pixels.

1.2 Limitation of Transductive Framework

However, there still two important issues exist in practical interactive image segmentation tasks. First, considering that the common interactive tools for user are broad brushes or region selection tools, it is hard to mark all the seeds without any mistakes by hand, especially when the foreground includes tiny object like hair, net, etc. Secondly, some prior statistical information implicitly provided by user input seeds is ignored in tranductive framework. Corresponding to these issues, we extend the tranductive framework by a novel method which is robust to user input fault as well as consistent with prior statistical information.

The rest of the article is organized as follows. Section 2 demonstrates our extension of transductive framework. Section 3 presents two algorithms based on our extension method with different loss function about the noise. Section 4 illustrates our experimental results. And Section 5 makes a brief conclusion.

2 Transductive Framework with Noise

From perspective of transducive framework, the interactive image segmentation problem is considered as a classification problem. The pixels that user supply as foreground or background seeds can be seen as labeled data, while others are unlabeled data. The goal of segmentation is to give foreground or background labels to all the pixels in the image. The principles of labeling the pixels are based on the following assumption:

- Although there are noises, most of the labels provided are correct. The final solution should change the previous labels as few as possible.
- The principles to classify the foreground and the background should be brief. In other words, if the principles are described as a function in some certain feature space, it should be smooth.
- If there exist some features that the pixels having the similar features are likely to have some labels, we can confirm that there exist some space where data nearby in it are likely belong to the same cluster. Data in the same cluster are likely to have the same label. Data with different labels are unlikely to appear in one cluster. If the principles are described as a function, its boundary is supposed to go across the area where data distribution density is relatively low.

2.1 Transductive Model

According to Vapnik's statistical learning theory [9], we formulate the segmentation problem and the assumptions as follows:

Let $(x_1, y_1) \ldots (x_l, y_l)$ be labeled data, where $Y_L = \{y_1 \ldots y_l\}$ are the class labels belonging to $\{1, -1\}$. Let $(x_{l+1}, y_{l+1}) \ldots (x_{l+u}, y_{l+u})$ be unlabeled data where $Y_U = \{ y_{l+1} \ldots y_{l+u} \}$ are unknown, $l + u = n$, usually $l << u$. Let $X = \{x_1 \ldots x_{l+u}\}$ where $x_i \in R^D$. The goal of segmentation is to estimate Y_U from X and Y_L by minimizing

$$f^* = \arg\min_{f \in H_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_K^2 + \gamma_I \| f \|_I^2 \tag{1}$$

where V is some loss function, H_K is an appropriately chosen Reproducing Kernel Hilbert Space of function $X \rightarrow R$ with corresponding norm $\| \|_K$, K is Gram matrix, H_I reflect the intrinsic structure of P_x.

The representer theorem in [12] states that the solution to this minimization problem exists in (1) only depends on the data points. Each minimizer of f meets the form

$$f^*(x) = \sum_{i=1}^{l} \alpha_i K(x_i, x) \tag{2}$$

Therefore, the problem is reduced to optimizing over the finite dimensional space of coefficients α_i.

2.2 Graph Laplacian Regulation

When the support of is a compact submanifold $M \subset X = R^n$. In that case, one natural choice for $\|f\|_I$ is $\int_M \langle \nabla_M f, \nabla_M f \rangle$, which can be approximated on the basis of labeled and unlabeled data using the graph Laplacian. Thus, the optimization problem becomes

$$f^* = \arg\min_{f \in H_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_K^2 + \frac{\gamma_I}{(u+l)^2} \sum_{i,j=1}^{l+u} (f(x_i) - f(x_j))^2 W_{ij}$$

$$= \arg\min_{f \in H_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_K^2 + \frac{\gamma_I}{(u+l)^2} \mathbf{f}^T L \mathbf{f} \tag{3}$$

where W_{ij} are edge weights in the data adjacency graph, $\mathbf{f} = (f(x_1), \ldots, f(x_{l+u}))^T$, L is the graph Laplacian given by $L = D - W$, and D is diagonal matrix containing elements $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. The normalizing coefficient $\frac{1}{(u+l)^2}$ is the natural scale factor for the empirical estimate of the Laplacian operator.

2.3 Combining with Prior Statistical Information

To enforce that all unlabeled data are labeled in proportion to the prior class probabilities implicitly provided by user seeds, we add the constraint

$$\frac{1}{l+u} \sum_{i=1}^{l+u} f(x_i) = \frac{1}{l} \sum_{i=1}^{l} y_i \tag{4}$$

which can also enforce balanced solution.

3 Implementation

We have presented the transductive framework with noise for interactive image segmentation. In the following discussion, we will present the kernel function, loss function and corresponding algorithm in the framework.

3.1 Kernel Selection

Lots of measures about features and kernels are used in image segmentation problems [2, 3, 4, 5]. The features such as intensity, grey histogram, RGB color, texture are all possible options, while the alternative kernels are even more. In this paper, we choose the simplest features: intensity and geometric position, and the typical RBF kernel. Even by these simple measures, we get accurate results.

Let $I(i)$ denote the image intensity at pixel i. Let $dist(i, j)$ denote the Euclidean distance between pixel i and j. We use the following kernel between pixels

$$K_{ij} = W_{ij} = dist(i, j)^{-1} e^{-\frac{(I(i)-I(j))^2}{o^2}}$$

We represent two extended regularization algorithms: Laplacian Regularized Least Squares(LaplacianRLS) and Laplacian Regularized SVM(LaplacianSVM) based on different loss function. The LaplacianRLS is easier and more efficient, while LaplacianSVM is more accurate which is proved in the experiments in Section 4.

3.2 Laplacian Regularized Least Squares

The Laplacian Regularized Least Squares algorithm solves problem (3) with the squared loss function:

$$\min_{f \in H_K} \frac{1}{l} \sum_{i=1}^{l} (y_i - f(x_i))^2 + \gamma_A \| f \|_K^2 + \frac{\gamma_I}{(u+l)^2} \mathrm{f}^T L \mathrm{f} \tag{5}$$

The Representer Theorem can be used to show that the solution is an expansion of kernel functions over both the labeled and the unlabeled data:

$$f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* K(x_i, x)$$

Substituting this form in the problem above, as before, we arrive at a convex differentiable objective function of the $l+u$ dimensional variable $\alpha = (\alpha_1, \ldots, \alpha_{l+u})^T$:

$$\alpha = \arg\min_{\alpha \in R^{l+u}} \frac{1}{l} (Y - JK\alpha)^T (Y - JK\alpha) + \gamma_A \alpha^T K \alpha + \frac{\gamma_I}{(u+l)^2} \alpha^T KLK\alpha \tag{6}$$

subject to the constraint from (4)

$$\frac{1}{l+u} E^T K\alpha - \frac{1}{l} E^T Y = 0 \tag{7}$$

where K is the $(l+u) \times (l+u)$ Gram matrix over labeled and unlabeled points; Y is an $(l+u)$ dimensional label vector given by $Y = (y_1, \ldots y_l, 0, \ldots 0)^T$; J is an $(l+u) \times (l+u)$ diagonal matrix with first l diagonal entries as 1 and the rest 0 and E_{l+u} is an $(l+u)$ dimensional vector with all the entries as 1.

The quadratic optimal problem (6)(7) can be solved easily by Lagrange multipliers technique.

3.3 Laplacian Support Vector Machines

By including the intrinsic smoothness penalty term, we can extend SVMs by solving the following problem:

$$\min_{f \in H_K} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(x_i))^2 + \gamma_A \| f \|_K^2 + \frac{\gamma_I}{(u+l)^2} \mathrm{f}^T L \mathrm{f} \tag{9}$$

Often in SVM formulations, an unregularized bias term b is added to the above form. Again, the primal problem can be easily seen to be the following

$$\min_{\alpha\in R^{l+u},\xi\in R^{l}} \frac{1}{l}\sum_{i=1}^{l}\xi_i + \gamma_A\alpha^T K\alpha + \frac{\gamma_I}{(u+l)^2}\alpha^T KLK\alpha \tag{10}$$

subject to

$$\begin{aligned} & y_i(\sum_{j=1}^{l+u}\alpha_j K(x_i,x_j)+b)\geq 1-\xi_i, \quad i=1,...,l \\ & \xi\geq 0, \qquad\qquad\qquad\qquad i=1,...,l \end{aligned} \tag{11}$$

With the help of Lagrange multipliers, we can get the dual problem of the problem (10)(11) [1] :

$$\alpha^* = P\beta^* \tag{12}$$

$$\beta^* = \arg\max_{\beta\in R^{l}} \sum_{i=1}^{l}\beta_i - \frac{1}{2}\beta^T Q\beta \tag{13}$$

subject to

$$\begin{aligned} & \sum_{i=1}^{l}\beta_i y_i = 0 \\ & 0\leq\beta_i\leq\frac{1}{l}, \quad i=1,...,l \\ & \frac{1}{l+u}E^T KP\beta - \frac{1}{l}E^T Y = 0 \end{aligned} \tag{14}$$

where

$$P = (2\gamma_A I + 2\frac{\gamma_I}{(u+l)^2}LK)^{-1}J^T Y$$

$$Q = YJKP$$

$J = (I\ 0)$ is an $l\times(l+u)$ matrix with I as the $l\times l$ identity matrix and $Y = \mathrm{diag}(y_1, \ldots, y_l)$.

The problem (13)(14) can be solved by standard SVM classifier. Substituting the solution into (12), we get the classifier of (9).

4 Experiments

We choose Berkley image database to perform our algorithm. In our experiments, we mark the user supplied seeds with about 10% noise. The parameters in our algorithm

are set as $\gamma_A l = 0.005$, $\gamma l = 0.05$. A part of the segmentation results is illustrated in Fig 2. In Table 1, we compare our two algorithms with Grabcut, Ramdom Walker, and transductive methods. The experimental results demonstrate that our method achieve higher accuracy. The results also indicate that the Laplacian SVM is more accurate than Laplacian LRS in our case.

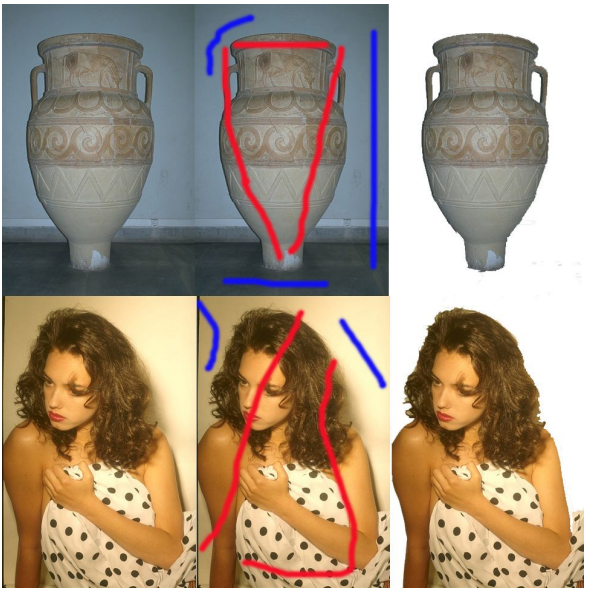


Fig. 2. Some results of our segmentation method

Table 1. Comparison of percentage of mislabeled pixels with user supplied seeds

Segmentation Method	Error Rate
Grabcut	6.4%
Random Walker	21.2%
Original Tranductive Method	17.1%
Laplacian LRS	5.1%
Laplacian SVM	4.7%

5 Conclusion

We propose a robust and accurate segmentation method in transductive framework. The method releases the requirement of user and is tolerant to a certain amount of user input faults as well as is combined with prior statistical information implicitly provided by user input seed. It is implemented by two different kinds of loss function. Experimental results demonstrate improvements in performance over the former approach.

References

1. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold Regulation: A Geometric Framework for Learning from examples. Technical Report, Uni. Chicago (2004)
2. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive Image Segmentation Using An Adaptive GMMRF Model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021. Springer, Heidelberg (2004)
3. Boykov, Y., Jolly, M.P.: Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In: ICCV (2001)
4. Duchenne, O., Audibert, J.Y., Keriven, R., Ponce, J., Segonne, F.: Segmentation by Transduction. In: CVPR (2008)
5. Grady, L.: Random Walks for Image Segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence (2006)
6. Rother, C., Kolmogorov, V., Blake, A.: GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts. In: SIGGRAPH (2004)
7. Xue, B., Sapiro, G.: A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In: ICCV (2007)
8. Zhu, X.: Semi-Supervised Learning Literature Survey. Technical Report, Uni. Wisconsin (2007)
9. Vapnik, V.: Statistical Learning Theory. Springer, Heidelberg (1998)
10. Schoelkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)

A New Approach of Feature Selection for Chinese Web Page Categorization

Cunhe Li, Lina Zhu, and Kangwei Liu

School of Computer & Communication Engineering,
China University of Petroleum, 257061, Dongying, China
jelly_3@163.com

Abstract. Feature selection is a key step of web page categorization. It can influence the accuracy of categorization directly as well as the efficiency. This paper proposes a new approach of feature selection based on Mutual Information algorithm. It brings in feature whose Mutual Information is negative and emphasizes the occurrence probabilities of features in different categories. Moreover, it makes some improvements on the web page preprocessing to reserve some useful features. The experiment shows that the new feature selection method improves the accuracy of categorization effectively.

Keywords: Web page categorization, Preprocessing, Feature selection, Mutual Information.

1 Introduction

With the great needs of information globalization and the rapid development of information technology, web page categorization has become one of the most attractive focuses in research. The web page categorization is a process using computers to classify large quantity of web pages automatically according to some categorization rules. It can organize the web pages orderly, improve the performance of Information Retrieval and increase the availability of web resources. At present, web page categorization has been widely used in search engine, information filter, digitized library and so on.

In Information Retrieval areas, Vector Space Model (VSM) is commonly used to index text information. So a web document can be denoted by vector $V(d)=(t_1, W_1(d), t_2, W_2(d), \ldots\ldots t_n, W_n(d))$ [1] where t_i is a feature and $W_i(d)$ is the weight of t_i. In a large-scale categorization system, the number of features is usually high to hundreds of thousands. However, high-dimension often leads to low accuracy and inefficiency because of the redundancy and the intractable computational complexity. Feature selection is a method to remove irrelevant features from high dimension space, it can reduce the need of memory, speed up text handling and improve precision for web page categorization [2]. It is an essential step of web page categorization which can influence the capability of categorization directly. Thus, how to select useful features among the high dimension dataset becomes an important research subject.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 386–395, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Among the common used feature selection methods, this paper mainly conducts a study on Mutual Information algorithm (MI), proposes some improvements toward its disadvantages. Meanwhile, it uses category special words to filter useless terms in the web pages preprocessing. Experiment is carried out on web datasets and the results show good performances of the new proposed feature selection method.

2 Web Page Preprocessing

As the web page is not a structural data type, it always contains HTML tags, hypertext links and other noises, so preprocessing is very necessary. That means to eliminate tags and marks from pages in order to transform them into text forms which can be handled easily. Considering Chinese words have no natural spaces between each other, we need to split the Chinese sentence into many single words, this preprocess is called "Chinese words segmentation" and it also has a distinct effect on categorization.

2.1 Primary Selection

In a large-scale categorization system, the number of terms after Chinese words segmentation is very great. If we take all these terms into feature subset by using VSM model, the dimension of vector space may be thousands upon thousands, even including some noisy features. This unsettled vector may cost much in the calculation process [3].

At the aspect of Chinese, nouns and verbs can represent category information effectively, and adjectives are usually used as modifiers. For empty words like prepositions and adverbs, they are meaningless for categorization [4]. Thus, we prefer to select nouns and verbs to be the primary feature subset.

2.2 Category Special Words

Among the primary feature subset, some terms may occur in all the categories, they're not representative and discriminating. Selecting them as features may make the borderline of category amphibolous. However, some terms occur in certain categories only. For example, "floating aerodrome" is quite likely to occur in "martial" category and "NBA" probably occurs in "sport" category. This kind of terms has extreme representation and discrimination which we name as "category special words". Unfortunately they're often removed due to their low occurrences in the other categories, with losing much available information. Therefore, we use the following method to reserve them and bring them into the feature subset:

Let $P(C_i \mid t_k) = \dfrac{P(C_i)P(t_k \mid C_i)}{P(t_k)}$ be the probability of each term t_k occurring in each category, and $P(t_k) = \sum_{i=1}^{m} P(C_i)P(t_k \mid C_i)$, where $P(C_i)$ is the probability of category C_i occurs in the whole categories. $P(t_k \mid C_i) = \dfrac{1 + \sum_{j=1}^{m_i} tf(t_{kj})}{W + \sum_{k=1}^{W}\sum_{j=1}^{m_i} tf(t_{kj})}$, where $tf(t_{kj})$ is the

time t_k occurs in the jth document of category C_i, m_i is the number of documents belonging to category C_i and W is the total number of all terms.

Expression $\Delta P(C_i \mid t_k) = \frac{Max_1}{Max_2}$ is used to measure the information that t_k provides to C_i, where Max_1 is the maximum of $P(C_i \mid t_k)$ and Max_2 is the second one. Choose the terms whose values are larger than the predetermined threshold T as "category special words". The larger the value is, the stronger the representation of t_k for C_i is.

3 Feature Selection Methods

Yang et al [5] conducted a comparative study on five current feature selection methods in text categorization, including document frequency (DF), mutual information (MI), information gain (IG), a χ^2-test (CHI) and term strength (TS). IG and CHI are found to be the most effective in their experiments, but MI is the worst. This paper analyses and improves the MI algorithm based on its disadvantages.

3.1 Mutual Information Algorithm

In information theory, mutual information measures the correlation between two signals. Similarly, MI measures the correlation between feature and category. Its expression is defined to be [6]:

$$MI(t_k, c_i) = \log \frac{P(t_k \mid c_i)}{P(t_k)}. \tag{1}$$

Where $P(t_k)$ is the probability of t_k occurs in entire documents, $P(t_k \mid C_i)$ is the probability of t_k occurs in category C_i. Apparently, MI measures the contribution of each term to each category according to the ratio: the frequency of t_k occurs in category C_i to the frequency of t_k occurs in the whole categories. The larger the ratio is, the bigger the contribution of term to category is.

3.2 Mentioned Problems

(1) Remove the features whose MI are negative

According to Eq.1, the value of MI is large when $P(t_k \mid C_i)$ is large and $P(t_k)$ is small. Otherwise, the value is small even negative when $P(t_k \mid C_i)$ is far less than $P(t_k)$. This means the feature probably occurs in other categories rather than category C_i. When we select the top N terms as the feature subset by using conventional MI Algorithm, the negative features sorted behind other higher MI scores features are always filtered. Actually, they are potential influence factors and as important as the features with large MI scores.

(2) Neglect the probabilities of terms occurring in certain categories
Providing there are terms t1and t2, their occurrence probabilities in all categories are 0.6、0.3 and 0.3、0.15 in category C_i . According to Eq.1, they gain a same MI score log0.5, implying that they have the same contribution to category C_i , but in fact t1 is more related to category C_i than t2 based on its probabilities in category C_i . Expending to high-dimension space, sometimes the same MI score may correspond to thousands of different terms, but the terms sorted behind may be deleted at random. Thus, the Eq.1 is unfair for category representation.

3.3 Improvements of the Algorithm

(1) Use the absolute value of MI
We use the absolute value of MI to overcome the first problem mentioned above. The improved expression is defined to be:

$$MI(t_k, c_i) = \left| \log \frac{P(t_k \mid c_i)}{P(t_k)} \right|. \tag{2}$$

With this improvement, the negative terms are placed former because their values become larger, so they can be reserved and selected into the feature subset, provide useful information to the categorization.

(2) Take into account the probability of terms occurring in some category
For the second problem, we improve the expression as follows:

$$MI(t_k, c_i) = P(t_k \mid c_i) \log \frac{P(t_k \mid c_i)}{P(t_k)}. \tag{3}$$

Using the example above, we obtain MI (t1) =0.3log0.5, MI (t2) =0.15log0.5 by Eq.3, so the contributions t1 and t2 provide to category C_i are distinguished. The improvement strengthens the correlation between feature and category, promotes the place of feature with available information.

(3) Final improvement of algorithm
We combine the two improvements above as the final improvement of MI algorithm and this is also the new approach of MI our paper proposes. It can be expressed as follows:

$$MI^*(t_k, c_i) = P(t_k \mid c_i) \left| \log \frac{P(t_k \mid c_i)}{P(t_k)} \right|. \tag{4}$$

This improvement not only avoids eliminating some useful terms, but also makes the category discrimination of features more obvious that can represent categories exactly.

4 Algorithm Process

The basic idea of our new improved method is: after noise eliminating and Chinese words segmentation, first select nouns and verbs as the primary feature subset. Then calculate the value of "category special words", filter features less than the predetermined threshold T. After sorting them based on the MI* scores calculated by Eq.4, select the top K terms to be the final feature vector and use them in the web pages categorization eventually (K is the optima value for categorization after times of adjustments).

The process of the improved feature selection algorithm is described as Fig.1 and the second step "web pages preprocessing" in Fig.1 includes four steps as follows:

(1) Transform web pages into text forms by removing all the tags and marks from them.

(2) Use ICTCLAS [7] of Chinese Academy of Science to segment Chinese words.

(3) Select nouns and verbs in texts by matching them with the regular expression ([^\\S!@#$%&*'A-Za-z0-9])+/ (n| v| vn)[4].

(4) Calculate $P(C_i \mid t_k)$ for each term, select terms whose values are larger than T as "category special words".

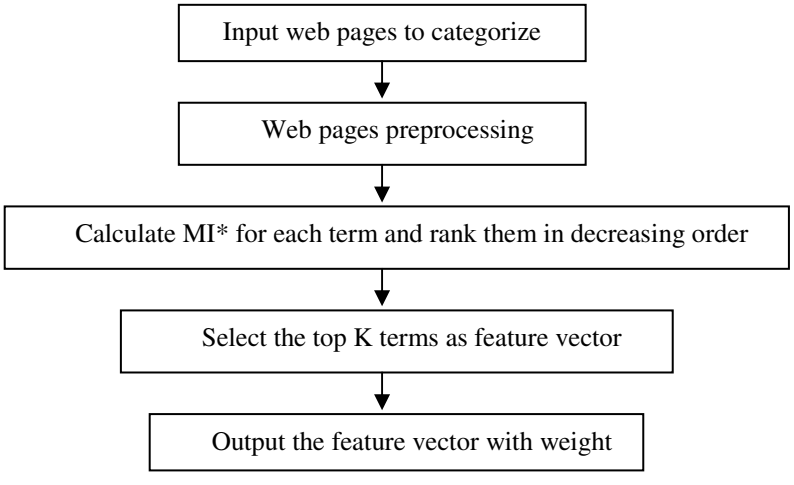


Fig. 1. The process of the feature selection algorithm

In the last step of Fig.1, the weight of feature vector is computed by TFIDF function which is frequently used in Text Retrieval and Machine Learning. Its formula is as follows:

$$W(t,\vec{d}) = \frac{tf(t,\vec{d}) \times \log(N/n_t + 0.01)}{\sqrt{\sum_{t \in \vec{d}} [tf(t,\vec{d}) \times \log(N/n_t + 0.01)]^2}}. \quad (5)$$

5 Experiments and Analysis

5.1 Experiment Data Set

As sina.com is an authoritative portal site and has exact categorization for web pages, so we choose it to gain the experiment data set and use "Web Crawler" to collect web pages starting from www.sina.com.cn. It contains 3170 training instances and 1630 testing instances which belong to 10 categories: News, Finance, Entertainment, Health, Sports, Education, Technology (IT), Real Estate, Military and Autos. Table 1 shows the number of web pages in each category both for training and testing.

Table 1. Numbers of each category for Training and Testing

No	Category	Training	Testing
1	News	340	190
2	Finance	320	170
3	Entertainment	280	120
4	Health	315	160
5	Sports	330	180
6	Education	296	150
7	Technology (IT)	330	165
8	Real Estate	283	130
9	Military	320	165
10	Autos	356	200
The total number of whole pages		3170	1630

5.2 Evaluation Criterion

In this paper, we use Macro-F1 [8] to evaluate the effectiveness of classifier. It is a measurement that combines recall and precision into a single score [9] according to the following formula:

$$Macro - F_1 = \frac{2 \times \sum_{i=1}^{m} p_i \times \sum_{i=1}^{m} r_i}{(\sum_{i=1}^{m} p_i + \sum_{i=1}^{m} r_i) \times m}. \tag{6}$$

Precision could be computed by the number of correctly categorized data over the number of all testing data. Recall could be computed by the number of correctly categorized data over the number of all the assigned data.

5.3 Results and Analysis

Classification algorithm is as important as the feature selection algorithm in the categorization system and the commonly used classification algorithm based on the VSM model are as the following: Support vector machine (SVM), K Nearest Neighbors

(KNN), Naive Bayes and neural networks. In experiment, we use the KNN algorithm to classify the web pages.

(1) We use different values of T to categorize web pages with the conventional method MI and the improved method MI* respectively, the effectiveness of categorization is as Fig.2.

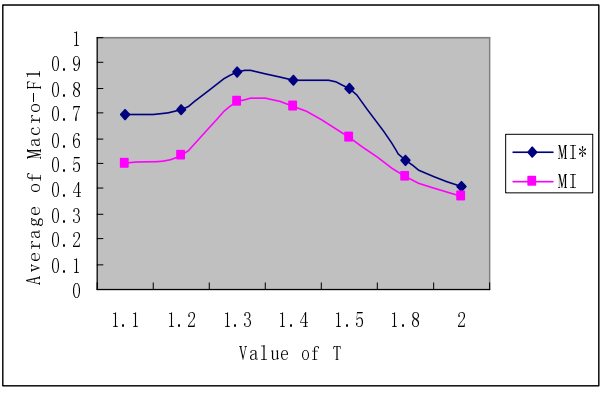


Fig. 2. Comparison of Macro-F1 with different T

Seeing from Fig.2, the Macro-F1 scores of two methods increase gradually with the increase of T and reach the peak values when T is 1.3, respectively Macro-F1 (MI) is 0.7478 and Macro-F1 (MI*) is 0.8658. It indicates that the accuracy of categorization can be improved by raising the value of T, but it tends to decline when T continues increase. This may because the discrimination required for category is so strict that the number of features becomes less as T increases, conversely, the category discrimination will decay while the number of features decreases. Considering both the category discrimination and the number of features selected, we choose T=1.3.

(2) For the dimension of web feature vector affects the accuracy and efficiency of categorization directly, we use the conventional method MI to test the effectiveness of classifier with different number of features.

Table 2. Comparison of Macro-F1 with different K

No	Feature dimension K	Macro-F1 Average
1	500	0.6503
2	800	0.6725
3	1000	0.6964
4	1500	0.7252
5	2000	0.7479
6	3000	0.7588
7	4000	0.7602

Table 2 demonstrates that Macro-F1 increases by more than 0.02 when feature dimension K is below 2000, but it descends to 0.0109 when K is 3000 and even less than 0.01 at 4000. So within a certain range, the effectiveness of categorization can be improved as the feature dimension increases, but it inclines to be stable while dimension exceeds a certain number. As the number of dimension can affect computational complexity obviously, we choose K=2000.

(3) The improved MI* algorithm is used to classify the 10 categories of web pages compared with the conventional MI algorithm. The result is shown in Table 3.

Table 3. Comparison of Macro-F1 with two methods

No	Category	MI	MI*
1	News	0.5493	0.5986
2	Finance	0.8258	0.8783
3	Entertainment	0.9126	0.9883
4	Health	0.7671	0.8675
5	Sports	0.8533	0.9582
6	Education	0.5982	0.6865
7	Technology (IT)	0.8719	0.9687
8	Real Estate	0.6652	0.8436
9	Military	0.7537	0.9795
10	Autos	0.6848	0.8899
Average of Macro-F1		0.74819	0.86591

We found that the Macro-F1 of MI* is obviously higher than that of MI and Fig.3 displays the increase percentage of Macro-F1 for each category. As we see, the Military category obtains the maximum, increasing by 22.58% while "Autos" get the second place with 20.51% increased. This is probably because there are more "category special words" in these two categories. Whereas "News" and "Finance" are at last mainly because they have many repeated features that are common in other

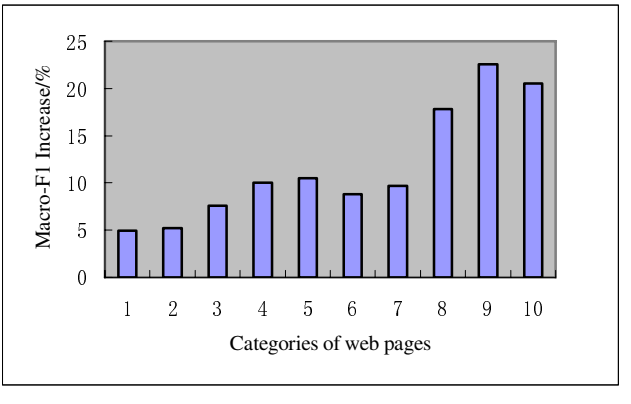


Fig. 3. Increase percentage of Macro-F1 of each category

categories. For instance, features in “Finance” or “Sports” usually belong to “News” and features in “Finance” may also include that of “Real Estate”.

(4) We also use IG and CHI methods to classify the 10 categories of web pages, comparing with the MI (*) algorithm. The result is shown as Table 4.

Table 4. Comparison of Macro-F1 with IG, CHI and MI (*)

No	Feature selection methods	Average of Macro-F1 / %
1	CHI	87.269
2	MI(*)	86.591
3	IG	86.457
4	MI	74.819

Compared with the most effective method (CHI, IG) mentioned above, the new proposed MI* algorithm improves the categorization accuracy obviously, it is close to CHI and even exceeds IG. Consequently, our experiments show that the improved Mutual Information algorithm is effective and the experimental results are quite good and reasonable.

6 Conclusions

This paper proposes a new approach of feature selection based on Mutual Information algorithm. It brings terms of negative MI into feature subset by using absolute value of MI and adds the occurrence probability of feature in some category to the conventional MI expression. Considering the incompleteness of stop-words list, we select nouns and verbs as the primary feature subset. Besides, we calculate “category special words” to avoid missing significative terms. Relative to the effective improvement of categorization accuracy, the additional computational complexity can be accepted.

During the feature selection, we only pay attention to the terms in titles and bodies, ignore the tags and hyperlinks of web pages which may contain some available information for categorization. So these elements need to be investigated in the future work. Furthermore, we will handle with adjectives which may influence the precision of web pages categorization in some way. In addition, this new proposed method will be tested on more data sets using some other classification algorithm.

References

1. Xu, J., Hu, M.: Feature Selection and Classification for Chinese Web Documents. Computer Engineering 31(8), 24–39 (2005)
2. Dash, M., Liu, H.: Feature Selection for Classification. International Journal of Intelligent Data Analysis 1, 131–156 (1997)
3. Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval 1(1/2), 67–88 (1999)

4. Hu, Y., Wu, H., Zhong, L.: Research of Feature Extraction Methods Based on Part of Speech in Chinese Documents Classification. Journal of WuHan University of Technology 29(4), 132–135 (2007)
5. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML 1997), pp. 412–420 (1997)
6. ZaffaIon, M., Hutter, M.: Robust Feature Selection by Mutual Information Distributions. In: Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence, UAI, pp. 577–584 (2002)
7. Zhang, H.: ICTCLAS 3.0 API of Chinese Academy of Science (2007), http://www.nlp.org.cn/project/project.php?proj_id=6
8. Li, X., Yan, H., Wang, J.: Search Engine Theory Technique and System 4(1), 189–190 (2005)
9. Kwak, N., Ho, C.: Input Feature Selection for Classification Problems. IEEE Transaction on Neural Network 13, 143–157 (2002)

Easily Reconfigurable Analytical Fuzzy Predictive Controllers: Actuator Faults Handling

Piotr M. Marusak

Institute of Control and Computation Engineering, Warsaw University of Technology,
Nowowiejska 15/19, 00–665 Warsaw, Poland
P.Marusak@ia.pw.edu.pl

Abstract. Efficient and easily reconfigurable predictive controllers are described in the paper. They are based on Takagi–Sugeno (TS) fuzzy models with local models in the form of the step responses. In these algorithms the TS fuzzy model is utilized in such a way that the control law can be calculated at each iteration in a simple and easy way. They are computationally efficient and can be easily reconfigured during adaptation to new situations, like e.g. actuator faults that can occur in the control system.

Keywords: Fuzzy control, predictive control, fault–tolerant control, nonlinear control, constrained control, Dynamic Matrix Control.

1 Introduction

Model predictive control (MPC) algorithms are formulated in such a way that all information about control system operation and on conditions in which it operates can be easily taken into consideration during the control signal derivation. This is the reason, why (MPC) algorithms are widely used in practice offering very good control performance even for control plants with difficult dynamics (inverse response, time delays) and for MIMO (Multiple Input Multiple Output) processes; see e.g. [1], [2], [4], [11], [14].

Standard formulations of the MPC algorithms are based on linear control plant models. An advantage of such an approach is that algorithms are formulated as easy to solve, quadratic optimization problems. Thus, in the unconstrained case, a control law can be easily obtained. Alas, application of the MPC algorithm based on a linear control plant model to a nonlinear plant may bring unsatisfactory results or the results can be improved using the algorithm based on a nonlinear model. It is especially important if control in a wide range of set point values is needed.

Unfortunately, direct usage of a nonlinear model to design the MPC algorithm leads to its formulation as a nonlinear, and in general, non–convex optimization problem that is hard to solve and computationally demanding. Easily reconfigurable analytical fuzzy controllers proposed in the paper are formulated in such a way that this problem is avoided. The controllers are based on Takagi–Sugeno (TS) fuzzy models [13] that are used in such a way that only the quadratic optimization problem must be solved in order to calculate the control signals.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 396–405, 2008.
© Springer-Verlag Berlin Heidelberg 2008

A very popular MPC algorithm applied in practice is the DMC algorithm; see e.g. [1], [2], [3], [4], [11], [14]. It uses step responses of the control plant as the model. Such a model can be obtained relatively easy. The controllers proposed in the paper are based on TS models with local models in the form of step responses. Thus, the design process of the proposed controller can be simplified to much extent. Usually the analytical fuzzy controllers are formulated using the PDC approach [12]. After application of this technique, the fuzzy DMC (FDMC) controller, composed of a few analytical DMC controllers, can be obtained [8]. However, such a controller is hard to reconfigure because the parameters of all local controllers must be either recalculated on–line or prepared off line and stored in the memory of the control device.

In the next section the proposed, easy to reconfigure, fuzzy analytical DMC controllers are formulated. It is shown that very simple control laws can be obtained for SISO (Single Input Single Output) and two–input two–output plants. In Sect. 3 mechanisms of actuator blockade toleration in the proposed algorithms are discussed and control laws obtained after application of these mechanisms are given. Sect. 4 contains description of a MIMO control plant (an evaporator), of controllers used during the experiments and discussion of obtained results. The results illustrate efficiency of the proposed mechanisms of actuator fault toleration, which can be easily introduced to the proposed FDMC controllers. The paper is shortly summarized in the last section.

2 Efficient Fuzzy Analytical Controllers

The proposed fuzzy controllers are based on TS models with local models in the form of step responses that are relatively easy to obtain. It is because it is sufficient to collect a few step responses of the control plant near a few operating points. The membership functions can be chosen using expert knowledge, simulation experiments, fuzzy neural networks or all these techniques combined. The FDMC controllers are thus based on TS fuzzy models described by the following rules:

$$\text{If } y_k^{j_y} \text{ is } B_1^{f,j_y} \text{ and} \ldots \text{and } y_{k-n_P+1}^{j_y} \text{ is } B_{n_P}^{f,j_y} \text{ and} \ldots \text{ and } u_k^{j_u} \text{ is } C_1^{f,j_u} \text{ and} \ldots \text{and } u_{k-m_P+1}^{j_u} \text{ is } C_{m_P}^{f,j_u}$$

$$\text{Then } \tilde{y}_{k+1}^{j,f} = \sum_{m=1}^{n_u} \sum_{i=1}^{p_d-1} a_i^{j,m,f} \cdot \Delta u_{k-i}^m + a_{p_d}^{j,m,f} \cdot u_{k-p_d}^m, \tag{1}$$

where $y_k^{j_y}$ is a value of the j_y^{th} output variable at the k^{th} sampling instant, $u_k^{j_u}$ is a value of the j_u^{th} manipulated variable at the k^{th} sampling instant, $B_1^{f,j_y},\ldots,B_{n_P}^{f,j_y}, C_1^{f,j_u},\ldots,C_{m_P}^{f,j_u}$ are fuzzy sets, $a_i^{j,m,f}$ (i=1,…,p_d) are the coefficients of step responses in the f^{th} local model describing influence of the m^{th} input on the j^{th} output, p_d is equal to the number of sampling instants after which the coefficients of the step responses can be assumed as settled, $j_y = 1,\ldots, n_y$, $j_u = 1,\ldots, n_u$, $f = 1,\ldots, l$, l is number of rules.

For current sampling instant using: current values of process variables, the TS model (1) and fuzzy reasoning, the following model is obtained (that is in fact the step response control plant model valid for the current values of process variables):

$$\breve{y}_k^j = \sum_{m=1}^{n_u} \sum_{i=1}^{p_d-1} \tilde{a}_i^{j,m} \cdot \Delta u_{k-i}^m + \tilde{a}_{p_d}^{j,m} \cdot u_{k-p_d}^m , \tag{2}$$

where $\breve{y}_k^j$ is the j^{th} output of the control plant model at the k^{th} sampling instant, $\tilde{a}_i^{j,m} = \sum_{f=1}^{l} \tilde{w}_f \cdot a_i^{j,m,f}$, and $\tilde{w}_f$ are the normalized weights; see e.g. [10], [13].

The output value for the $(k+i)^{\text{th}}$ sampling instant predicted at the k^{th} sampling instant for the j^{th} output is then calculated using the following formula:

$$y_{k+i|k}^j = \sum_{m=1}^{n_u} \left(\sum_{n=1}^{i} \tilde{a}_n^{j,m} \cdot \Delta u_{k-n+i}^m + \sum_{n=i+1}^{p_d-1} \tilde{a}_n^{j,m} \cdot \Delta u_{k-n+i}^m + \tilde{a}_{p_d}^{j,m} \cdot u_{k-p_d+i}^m \right) + d_k^j , \tag{3}$$

where $d_k^j = y_k^j - \breve{y}_{k-1}^j$ is assumed to be the same at each sampling instant in the prediction horizon (the DMC–type disturbance model). Thus, assuming that each manipulated variable can change only once during the prediction horizon, (3) can be transformed into the following form:

$$\begin{aligned} y_{k+i|k}^j = y_k^j + \sum_{m=1}^{n_u} \left(\sum_{n=i+1}^{p_d-1} \tilde{a}_n^{j,m} \cdot \Delta u_{k-n+i}^m + \tilde{a}_{p_d}^{j,m} \cdot \sum_{n=p_d}^{p_d+i-1} \Delta u_{k-n+i}^m - \sum_{n=1}^{p_d-1} \tilde{a}_n^{j,m} \cdot \Delta u_{k-n}^m \right) \\ + \sum_{m=1}^{n_u} \tilde{a}_i^{j,m} \cdot \Delta u_{k|k}^m = \tilde{y}_{k+i|k}^j + \sum_{m=1}^{n_u} \tilde{a}_i^{j,m} \cdot \Delta u_{k|k}^m , \end{aligned} \tag{4}$$

where only the last component depends on future manipulated variable changes $\Delta u_{k|k}^j$, $\tilde{y}_{k+i|k}^j$ are elements of, so called, free response of the plant that describe influence of the control signals applied to the control plant in the past on control plant outputs.

Using the prediction (4) one can formulate the optimization problem:

$$\min_{\Delta \boldsymbol{u}} \sum_{j=1}^{n_y} \sum_{i=1}^{p} \kappa_j \cdot \left(\bar{y}_k^j - y_{k+i|k}^j \right)^2 + \sum_{j=1}^{n_u} \lambda_j \cdot \left(\Delta u_{k|k}^j \right)^2 , \tag{5}$$

where $\bar{y}_k^j$ is a set–point value for j^{th} output (as in most cases, constant set–points on the whole prediction horizon are assumed), $\kappa_j \geq 0$ and $\lambda_j \geq 0$ are weighting coefficients for the predicted control errors of the j^{th} output and for the changes of the j^{th} manipulated variable, respectively, p denotes prediction horizon, n_y, n_u denote number of output and manipulated variables, respectively, $\Delta \boldsymbol{u} = \left[\Delta u_{k|k}^1, \ldots, \Delta u_{k|k}^{n_u} \right]^T$ is the vector of calculated future changes of manipulated variables. The vector of predicted output values $\boldsymbol{y} = \left[\boldsymbol{y}_k^1, \boldsymbol{y}_k^2, \ldots, \boldsymbol{y}_k^{n_y} \right]^{\mathrm{T}}$, $\boldsymbol{y}_k^j = \left[y_{k+1|k}^j, \ldots, y_{k+p|k}^j \right]$, can be decomposed, according to (4), as follows:

$$\boldsymbol{y} = \tilde{\boldsymbol{y}} + \boldsymbol{A} \cdot \Delta \boldsymbol{u}, \tag{6}$$

where $\tilde{\boldsymbol{y}} = \left[\tilde{\boldsymbol{y}}_k^1, \tilde{\boldsymbol{y}}_k^2, \ldots, \tilde{\boldsymbol{y}}_k^{n_y}\right]^{\mathrm{T}}$, $\tilde{\boldsymbol{y}}_k^j = \left[\tilde{y}_{k+1|k}^j, \ldots, \tilde{y}_{k+p|k}^j\right]$ is the free response of the control plant, $\boldsymbol{A}$ is a $(p \cdot n_y) \times n_u$ matrix, called the dynamic matrix and composed of coefficients of the control plant step response

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \cdots & \boldsymbol{A}_{1n_u} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \cdots & \boldsymbol{A}_{2n_u} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n_y 1} & \boldsymbol{A}_{n_y 2} & \cdots & \boldsymbol{A}_{n_y n_u} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{A}_{mn} = \left[\tilde{a}_1^{m,n} \quad \tilde{a}_2^{m,n} \quad \cdots \quad \tilde{a}_p^{m,n}\right]^T$.

The optimization problem (5) can be written in the vector form as:

$$\min_{\Delta \boldsymbol{u}} \left[\left(\boldsymbol{A} \cdot \Delta \boldsymbol{u} - (\bar{\boldsymbol{y}} - \tilde{\boldsymbol{y}})\right)^T \cdot \boldsymbol{\kappa} \cdot \left(\boldsymbol{A} \cdot \Delta \boldsymbol{u} - (\bar{\boldsymbol{y}} - \tilde{\boldsymbol{y}})\right) + \Delta \boldsymbol{u}^T \cdot \boldsymbol{\lambda} \cdot \Delta \boldsymbol{u} \right], \tag{8}$$

where $\bar{\boldsymbol{y}} = \left[\bar{\boldsymbol{y}}_k^1, \bar{\boldsymbol{y}}_k^2, \ldots, \bar{\boldsymbol{y}}_k^{n_y}\right]^{\mathrm{T}}$, $\bar{\boldsymbol{y}}_k^j = \left[\bar{y}_k^j, \ldots, \bar{y}_k^j\right]$, $\boldsymbol{\kappa} = \left[\kappa_1, \ldots, \kappa_{n_y}\right] \cdot \boldsymbol{I}$, $\boldsymbol{\kappa}_j = \left[\kappa_j, \ldots, \kappa_j\right]$, $\boldsymbol{\lambda} = \left[\lambda_1, \ldots, \lambda_{n_u}\right] \cdot \boldsymbol{I}$, are matrices of appropriate dimensions, $\boldsymbol{I}$ is the identity matrix.

If the problem (8) is minimized without constraints, then an analytical solution can be found using the following formula; see e.g. in [14]:

$$\Delta \boldsymbol{u} = \left(\boldsymbol{A}^T \cdot \boldsymbol{\kappa} \cdot \boldsymbol{A} + \boldsymbol{\lambda}\right)^{-1} \cdot \boldsymbol{A}^T \cdot \boldsymbol{\kappa} \cdot \left(\bar{\boldsymbol{y}} - \tilde{\boldsymbol{y}}\right). \tag{9}$$

In the case when the number of outputs and inputs of the control plant is relatively small, which is true in most cases, the formula (9) is simple and easy to calculate. Let us now consider two cases.

2.1 The Case of a SISO Control Plant

The simplest is the case of the SISO plant, when n_y=1 and n_u=1. Then:

$$\boldsymbol{A} = \boldsymbol{A}_{11} = \left[\tilde{a}_1^{1,1} \quad \tilde{a}_2^{1,1} \quad \cdots \quad \tilde{a}_p^{1,1}\right]^T = \left[\tilde{a}_1 \quad \tilde{a}_2 \quad \cdots \quad \tilde{a}_p\right]^T, \tag{10}$$

$$\boldsymbol{\lambda} = \lambda_1 = \lambda, \tag{11}$$

$$\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}}_k^1 = \left[\bar{y}_k^1, \ldots, \bar{y}_k^1\right] = \left[\bar{y}_k, \ldots, \bar{y}_k\right], \tag{12}$$

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{y}}_k^1 = \left[\tilde{y}_{k+1|k}^1, \ldots, \tilde{y}_{k+p|k}^1\right] = \left[\tilde{y}_{k+1|k}, \ldots, \tilde{y}_{k+p|k}\right]. \tag{13}$$

Without loss of generality one can assume that κ_1=1. Thus, the change in the manipulated variable can be obtained using the following simple and easy to calculate formula:

$$\Delta u_{k|k} = \frac{\sum_{i=1}^{p} \tilde{a}_i \cdot \left(\bar{y}_k - \tilde{y}_{k+i|k}\right)}{\sum_{i=1}^{p} \left(\tilde{a}_i\right)^2 + \lambda}. \tag{14}$$

2.2 The Case of a Control Plant with Two Inputs and Two Outputs

Most of MIMO control systems described in the literature are systems designed for processes with two inputs and two outputs. Let us now consider this case. Let $\boldsymbol{K} = \boldsymbol{A}^T \cdot \boldsymbol{\kappa} \cdot \boldsymbol{A} + \boldsymbol{\lambda}$ then

$$\boldsymbol{K} = \begin{bmatrix} \sum_{i=1}^{p}\left(\kappa_1\left(a_i^{1,1}\right)^2 + \kappa_2\left(a_i^{2,1}\right)^2\right) + \lambda_1 & \sum_{i=1}^{p}\left(\kappa_1 a_i^{1,1}a_i^{1,2} + \kappa_2 a_i^{2,1}a_i^{2,2}\right) \\ \sum_{i=1}^{p}\left(\kappa_1 a_i^{1,1}a_i^{1,2} + \kappa_2 a_i^{2,1}a_i^{2,2}\right) & \sum_{i=1}^{p}\left(\kappa_1\left(a_i^{1,2}\right)^2 + \kappa_2\left(a_i^{2,2}\right)^2\right) + \lambda_2 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}. \tag{15}$$

Thus

$$\Delta\boldsymbol{u} = \begin{bmatrix} \Delta u_{k|k}^1 \\ \Delta u_{k|k}^2 \end{bmatrix} = \frac{1}{k_{11}k_{22} - k_{12}k_{21}} \begin{bmatrix} k_{22} & -k_{12} \\ -k_{21} & k_{11} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{p}\left(\kappa_1 \cdot \tilde{a}_i^{1,1} \cdot \left(\bar{y}_k^1 - \tilde{y}_{k+i|k}^1\right) + \kappa_2 \cdot \tilde{a}_i^{2,1} \cdot \left(\bar{y}_k^2 - \tilde{y}_{k+i|k}^2\right)\right) \\ \sum_{i=1}^{p}\left(\kappa_1 \cdot \tilde{a}_i^{1,2} \cdot \left(\bar{y}_k^1 - \tilde{y}_{k+i|k}^1\right) + \kappa_2 \cdot \tilde{a}_i^{2,2} \cdot \left(\bar{y}_k^2 - \tilde{y}_{k+i|k}^2\right)\right) \end{bmatrix}. \tag{16}$$

For systems with more inputs and outputs the solution (9) can be obtained using numerical procedure to inverse the $\boldsymbol{K}$ matrix.

Remark. In order to simplify calculation of the manipulated variable even more, one can take into consideration, in the performance index of the problem (5) control errors only at the end of the prediction horizon. However, in such a case very careful choice of prediction horizon is needed.

2.3 Manipulated Variable Constraints

The constraints put on manipulated variables, existing in the control system should be taken into consideration during calculation of values of the manipulated variables, otherwise control system performance will deteriorate. In the case of analytical predictive controllers a suitable mechanism is relatively simple and consists in application of the following rules for constraints put on (see e.g. [14]):

a) manipulated variable changes:

– if $\Delta u_{k|k}^j < \Delta u_{\min}^j$, then $\Delta u_{k|k}^j = \Delta u_{\min}^j$,

– if $\Delta u_{k|k}^j > \Delta u_{\max}^j$, then $\Delta u_{k|k}^j = \Delta u_{\max}^j$;

b) manipulated variable values:

– if $u_{k-1}^j + \Delta u_{k|k}^j < u_{\min}^j$, then $\Delta u_{k|k}^j = u_{\min}^j - u_{k-1}^j$,

– if $u_{k-1}^j + \Delta u_{k|k}^j > u_{\max}^j$, then $\Delta u_{k|k}^j = u_{\max}^j - u_{k-1}^j$.

3 Actuator Blockade Handling

Modification of the proposed fuzzy predictive controllers after an actuator blockade is the subject of this section. The method used in the paper was designed for the case when the goal of the control system was to continue operation after the failure occurrence in

as efficient way as possible, till the failure is fixed. The method consists in usage of the control plant model with eliminated input that was affected by the actuator blockade [7]. It will be now adapted to the algorithms proposed in the paper. The modification is relatively simple and consists in elimination, from the matrix $\boldsymbol{A}$ from (7), of the columns describing dependence of output variables on the manipulated variable affected by the failure:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1b} & \cdots & \boldsymbol{A}_{1n_u} \\ \boldsymbol{A}_{21} & \cdots & \boldsymbol{A}_{2b} & \cdots & \boldsymbol{A}_{2n_u} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n_y 1} & \cdots & \boldsymbol{A}_{n_y b} & \cdots & \boldsymbol{A}_{n_y n_u} \end{bmatrix}, \tag{17}$$

where b is the number of the manipulated variable affected by the actuator blockade. The vector of future control changes is now as follows:

$$\Delta \boldsymbol{u} = \left[\Delta u_{k|k}^{1}, \ldots, \Delta u_{k|k}^{b-1}, \Delta u_{k|k}^{b+1}, \ldots, \Delta u_{k|k}^{n_u}\right]^T. \tag{18}$$

Moreover, the weighting matrix $\boldsymbol{\lambda}$ is modified:

$$\boldsymbol{\lambda} = \left[\lambda_1, \ldots, \lambda_{b-1}, \lambda_{b+1}, \ldots, \lambda_{n_u}\right] \cdot \boldsymbol{I}. \tag{19}$$

Thus, the performance index (5) and the solution (9) change. In the case of a control plant with two inputs and two outputs, the following, very simple, solution is obtained:

$$\boldsymbol{K} = \left[\sum_{i=1}^{p}\left(\kappa_1\left(a_i^{1,na}\right)^2 + \kappa_2\left(a_i^{2,na}\right)^2\right) + \lambda_{na}\right] = k. \tag{20}$$

Then

$$\Delta u_{k|k}^{na} = \frac{\sum_{i=1}^{p}\left(\kappa_1 \cdot \tilde{a}_i^{1,na} \cdot \left(\bar{y}_k^1 - \tilde{y}_{k+i|k}^1\right) + \kappa_2 \cdot \tilde{a}_i^{2,na} \cdot \left(\bar{y}_k^2 - \tilde{y}_{k+i|k}^2\right)\right)}{k}. \tag{21}$$

where *na* is the index of a manipulated variable that is not affected by the fault.

If before the actuator blockade the number of manipulated variables was the same as the number of output variables (a typical case) then after the blockade of an actuator it is not possible to control all output variables anymore. The problem is especially important if constraints put on values of an output variable must be satisfied. It should be, however, stressed that analytical predictive controllers do not have efficient mechanisms of output constraints handling. They are designed as the controllers acting in the basic feedback layer in the multilayer control system structure. Thus, the output constraint fulfillment should be the task for either the constraint control layer in which appropriately designed controllers may be used or for the steady–state optimization layer [5], [6], [7], [14]. In both cases the set–points for the analytical controller are appropriately generated.

It is, however, advisable to use a mechanism that can positively influence the satisfaction of constraints by better stabilization of the chosen output (usually at the cost

of increase of control errors of other outputs). The standard way to do this is to use the weights κ_j from the performance index (5). Increasing the value of the κ_j weight causes better stabilization of the j^{th} output. There is, however, a problem how to choose values of the κ_j parameters.

The other solution is to eliminate from the performance index (5) predicted control errors of all outputs that need not be precisely stabilized (it can be interpreted as assigning 0 to the κ_j weights of the outputs eliminated from the performance index). After this modification simplified analytical controllers are obtained.

For further considerations it is assumed that the g^{th} output is eliminated. In such a case, rows describing dependence of the g^{th} output variable on the manipulated variables are eliminated from the matrix $\boldsymbol{A}$ [7]:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \cdots & \boldsymbol{A}_{1n_u} \\ \vdots & \vdots & \ddots & \vdots \\ \cancel{\boldsymbol{A}_{g1}} & \cancel{\boldsymbol{A}_{g2}} & \cancel{\cdots} & \cancel{\boldsymbol{A}_{gn_u}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n_y 1} & \boldsymbol{A}_{n_y 2} & \cdots & \boldsymbol{A}_{n_y n_u} \end{bmatrix}. \tag{22}$$

Moreover, the vectors $\bar{\boldsymbol{y}}$, $\tilde{\boldsymbol{y}}$ and matrix $\boldsymbol{\kappa}$ are modified:

$$\bar{\boldsymbol{y}} = \left[\bar{\boldsymbol{y}}_k^1, \ldots, \bar{\boldsymbol{y}}_k^{g-1}, \bar{\boldsymbol{y}}_k^{g+1}, \ldots, \bar{\boldsymbol{y}}_k^{n_y}\right]^{\mathrm{T}}, \tag{23}$$

$$\tilde{\boldsymbol{y}} = \left[\tilde{\boldsymbol{y}}_k^1, \ldots, \tilde{\boldsymbol{y}}_k^{g-1}, \tilde{\boldsymbol{y}}_k^{g+1}, \ldots, \tilde{\boldsymbol{y}}_k^{n_y}\right]^{\mathrm{T}}, \tag{24}$$

$$\boldsymbol{\kappa} = \left[\boldsymbol{\kappa}_1, \ldots, \boldsymbol{\kappa}_{g-1}, \boldsymbol{\kappa}_{g+1}, \ldots, \boldsymbol{\kappa}_{n_y}\right] \cdot \boldsymbol{I}. \tag{25}$$

Let us demonstrate how simple will be the control law in the case when a control plant has two inputs and two outputs. In such a case

$$\boldsymbol{K} = \left[\sum_{i=1}^{p}\left(\kappa_{ne}\left(a_i^{ne,na}\right)^2\right) + \lambda_{na}\right] = k, \tag{26}$$

thus

$$\Delta u_{k|k}^{na} = \frac{\sum_{i=1}^{p}\left(\kappa_{ne} \cdot \tilde{a}_i^{1,na} \cdot \left(\bar{y}_k^{ne} - \tilde{y}_{k+i|k}^{ne}\right)\right)}{k}. \tag{27}$$

where *ne* is the index of an output variable that should be better stabilized (is not eliminated from the performance index).

Remark 1. For the case of a control plant with two inputs and two outputs (majority of MIMO plants discussed in the literature), the obtained control laws are extremely simple. After making the dynamic matrix smaller the problem becomes in fact a problem with single input.

Remark 2. It is also possible to calculate the free response using TS models with other control plant models, like state space or difference equations. It should be,

however, noticed that all presented formulas, will be still valid because the dynamic matrix is present also in algorithms based on other control plant models [14].

4 Simulation Example

The control plant is an evaporator described in [9]. The output variables of the process are: $L2$ – level of the liquid in the separator (stabilized near $\overline{L}2 = 1$ m by a PI controller with parameters $K_p = 5.6$ and $T_i = 8.84$ min, proposed in [9]), $X2$ – product composition, $P2$ – pressure in the evaporator. The manipulated variables are: $F2$ – product flow (variable used to stabilize $L2$ level), $P100$ – steam pressure, $F200$ – cooling water flow. The diagram of the control plant is shown in Fig. 1a.

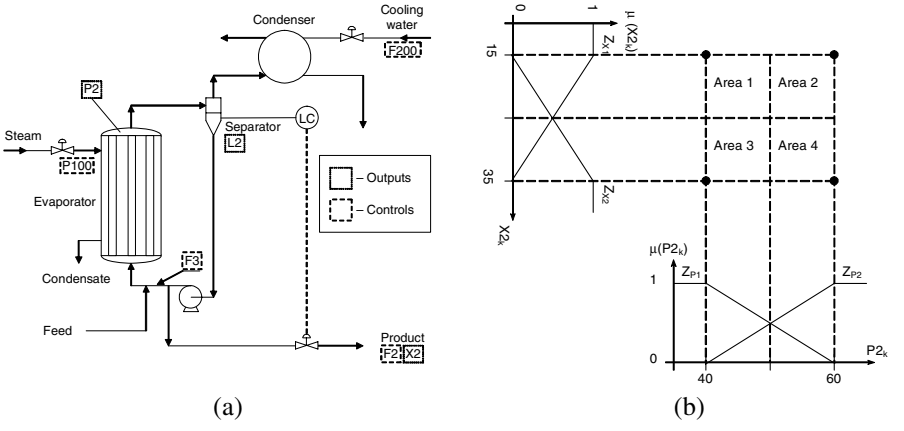


Fig. 1. a) Diagram of the evaporator installation; b) Membership functions of the FDMC controller

For the control plant an analytical FDMC algorithm was designed using step responses obtained from environs of four operating points that are shown in Fig. 1b together with assumed membership functions. From the point of view of the FDMC controller, the manipulated variables are: steam pressure $P100$ and cooling water flow $F200$; the controlled variables are: product composition $X2$ and pressure in the evaporator $P2$. The following values of controller parameters were assumed: $\kappa_{P2}=\kappa_{X2}=1$, $\lambda_{P100}=\lambda_{F200}=1$, $p=50$.

It was also assumed that the product composition is constrained (customer demand) – $X2_{min}$=25 %. Both manipulated variables are also constrained and $P100_{min}$=0 kPa, $P100_{max}$=400 kPa, $F200_{min}$=0 kg/min, $F200_{max}$=400 kg/min [9].

In Fig. 2 there are shown responses obtained in the control system with FDMC controller. It was assumed that blockade of the $F200$ actuator occurred at the beginning of the experiment and that this information is available (e.g. provided by an auto–diagnosing actuator). The reconfiguration was done after 5 minutes. In the 150th minute disturbance $X1$ (composition of the raw substance) changed from $X1_0$=5% to $X1_0$=4.7%.

After detection of the fault the dynamic matrix was modified and the control law calculated using (21). Moreover, the set–point value of the product composition $X2$ was set to 26% in order to shift the $X2$ variable away the constraint. Unfortunately, after the change of the disturbance, the product composition $X2$ approached very closely the constraint and violated it at the end of the experiment (dotted lines in Fig. 2). Thus, the future control errors of pressure $P2$ were eliminated from the controller performance index (the control law (27) was used). After this modification of the controller, the product composition $X2$ settles on the set–point value and is far from the constraint, even after the change of the $X1$ raw substance composition (dashed lines in Fig. 2). Next, the λ_{P100} parameter was changed to 5. As a result, better responses were obtained (solid lines in Fig. 2): the settling time is now smaller as well as the overshoot after the change of the $X2$ set–point value.

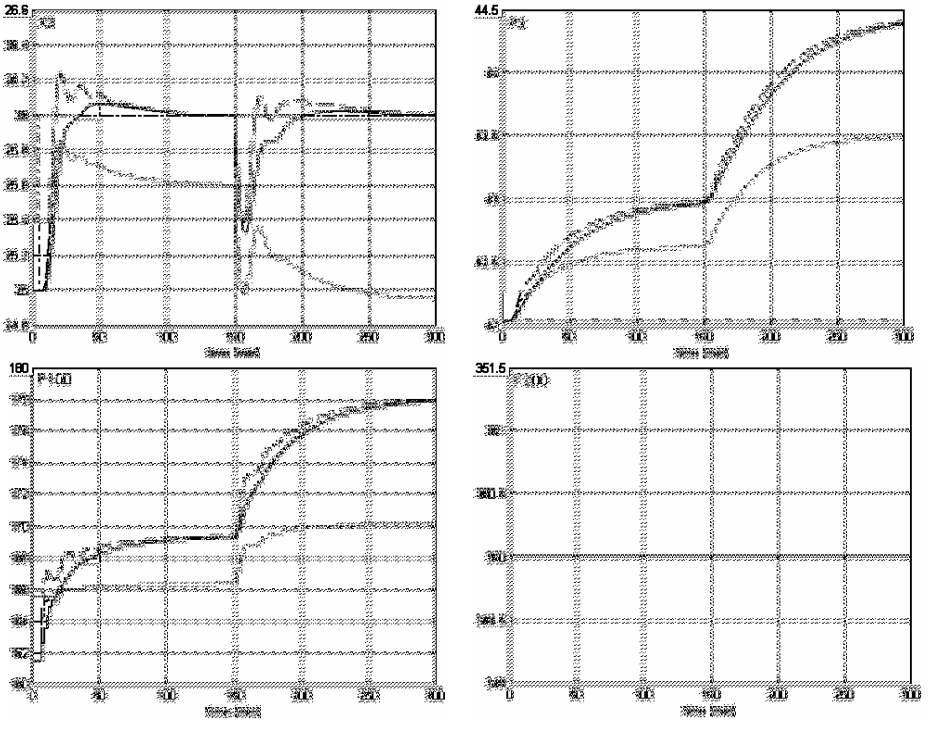


Fig. 2. Responses of the control system with an analytical FDMC controller; blockade of the F200 actuator taken into consideration and changes in the algorithm performance index made: $\kappa_{P2} = 1$ – dotted line, $\kappa_{P2} = 0$ – dashed line, $\kappa_{P2} = 0$ and $\lambda_{P100} = 5$ – solid line; above – output variables: X2 and P2, below – manipulated variables: P100 and F200

5 Summary

The fuzzy predictive controllers proposed in the paper are formulated in such a way that the control law can be obtained analytically. The controllers can be successfully

applied in control structures designed for fast processes, e.g. where they act in the basic feedback control layer and constraints are handled by the controllers acting in the constraint control layer.

Thanks to easiness of reconfiguration of the proposed controllers they are useful when modification of the control law is needed, e.g. after fault occurrence. It was demonstrated how to do such modification in response to actuator faults that can occur in the system. It was shown that very simple form of the control law is obtained after application of the proposed mechanisms of actuator blockade handling, for control plants with two inputs and two outputs. Usefulness of the proposed ideas was demonstrated in the control system of the nonlinear MIMO control plant in which the analytical FDMC controller was successfully applied and reconfigured after actuator blockade.

References

1. Blevins, T.L., McMillan, G.K., Wojsznis, W.K., Brown, M.W.: Advanced Control Unleashed. ISA (2003)
2. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, Heidelberg (1999)
3. Cutler, C.R., Ramaker, B.L.: Dynamic Matrix Control – a computer control algorithm. In: Proc. Joint Automatic Control Conference, San Francisco, CA, USA (1979)
4. Maciejowski, J.M.: Predictive control with constraints. Prentice Hall, Harlow (2002)
5. Ławryńczuk, M., Marusak, P., Tatjewski, P.: Set–point optimisation and predictive constrained control for fast feedback controlled processes. In: Proc. 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics MMAR 2007, Szczecin, Poland, pp. 357–362 (2007)
6. Marusak, P.: Efficient fuzzy predictive economic set–point optimizer. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS, vol. 5097, pp. 273–284. Springer, Heidelberg (2008)
7. Marusak, P.: Actuator fault toleration in control systems with analytical predictive controllers and output constraints. In: Proc. 13th IEEE/IFAC International Conference MMAR 2007, Szczecin, Poland, pp. 825–832 (2007)
8. Marusak, P., Tatjewski, P.: Stability analysis of nonlinear control systems with unconstrained fuzzy predictive controllers. Archives of Control Sciences 12(3), 267–288 (2002)
9. Newell, R.B., Lee, P.L.: Applied process control – a case study. Prentice Hall, Englewood Cliffs (1989)
10. Piegat, A.: Fuzzy Modeling and Control. Physica–Verlag, Berlin (2001)
11. Rossiter, J.A.: Model–Based Predictive Control. CRC Press, Boca Raton (2003)
12. Sala, A., Guerra, T.M., Babuska, R.: Perspectives of fuzzy systems and control. Fuzzy Sets and Systems 156, 432–444 (2005)
13. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Trans. on Systems, Man and Cybernetics 15, 116–132 (1985)
14. Tatjewski, P.: Advanced Control of Industrial Processes; Structures and Algorithms. Springer, London (2007)

Incremental Manifold Learning Algorithm Using PCA on Overlapping Local Neighborhoods for Dimensionality Reduction

Yubin Zhan, Jianping Yin, Guomin Zhang, and En Zhu

School of Computer, National University of Defense Technology, Changsha, Hunan, China
zhanyubin_dm@yahoo.com.cn

Abstract. A novel manifold learning algorithm called LPcaML is proposed in this paper. Based on the geometric intuition that d-dimensional manifold locally lies on or close to d-dimensional linear space, LPcaML first finds an α-TSLN of the whole high-dimensional input data set and then obtains the low-dimensional local coordinates of each neighborhood in the α-TSLN using classical PCA technique while preserving the local geometric and topological property of each neighborhood. At last LPcaML transforms each local coordinates to a unified global low-dimensional representation by processing each neighborhood in their order appeared in α-TSLN. And the transformation function of each neighborhood is obtained by solving a least square problem via the overlapped examples. By using the divide and conquer strategy, LPcaML can learn from incremental data and discover the underlying manifold efficiently even if the data set is large scale. Experiments on both synthetic data sets and real face data sets demonstrate the effectiveness of our LPcaML algorithm. Moreover the proposed LPcaML can discover the manifold from sparsely sampled data sets where other manifold learning algorithms can't.

Keywords: dimensionality reduction, incremental learning, least squares fit, local PCA.

1 Introduction

Dimensionality reduction has attracted extensive attentions. It is an effective way to conquer the curse of dimensionality[1]. There are many linear dimensionality reduction techniques such as Principal Component Analysis (PCA)[2], Multidimensional Scaling(MDS)[3]. However, these linear methods can't learn the nonlinear structure of nonlinear high-dimensional data, so various nonlinear extensions have been proposed, such as Self-organizing maps(SOM)[4], principal curves[5]. But these methods still have some difficulties while dealing with the high-dimensional nonlinear data[6].

After that, three articles about manifold were published on the same issue of Science. One discussed the manifold ways of human perception[7], the other two proposed two manifold learning algorithms: Isometric Mapping(ISOMAP)[8] and Locally Linear Embedding(LLE)[9]. Both the two nonlinear dimensionality reduction techniques are derived from a novel assumption that the high-dimensional input data lie on or close to a smooth low-dimensional manifold. Then based on this assumption

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 406–415, 2008.
© Springer-Verlag Berlin Heidelberg 2008

many new manifold learning algorithms were proposed, such as: Laplacian eigenmaps[10], LTSA[11]. These algorithms obtain low-dimensional embeddings by preserving different geometry properties of the underlying manifold.

Although above algorithms can achieve good embedding results on some data sets, they still have some problems: 1. Existing algorithms require that the underlying low-dimensional manifold is isometric (or conformal) to a subset of Euclidean Space[12]. 2. Global optimization problem of large scale dataset. Most of the existing approaches, whether global or local, reduce the problem to a global optimization problem which is very challenging to solve. 3. Incremental data problem. Most former methods operate in "batch" mode and get the low-dimensional embedded coordinates of all data points simultaneously. So it is hard for them to get the low-dimensional embedded coordinates of the new incremental data. Although some algorithms such as [13, 14]begin to focus on the incremental data, they still have some limitations.

The latter two problems are caused by a requirement of getting the low-dimensional embedded coordinates of the given data simultaneously. One possible way to solve this problem is to get the low-dimensional embedded coordinates in batches. When data are processed in batches, it is easy to preserve the local properties of the dataset. The nature of manifold is that it can be locally endowed coordinates, that is to say in every local patch (every point neighborhood) d-dimensional manifold lies on or close to d-dimensional linear space. So for each neighborhood, we can get low-dimensional representation by using classical linear dimensionality reduction techniques such as PCA on it while preserving geometric and topological property of the neighborhood. Therefore, based on this geometric intuition, this paper proposes a novel manifold learning algorithm called LPcaML. LPcaML algorithm works as follows: first it finds a Traversal Sequence of Local Neighborhood with overlapping factor α(α-TSLN) of input data set, and then for every local neighborhood in the α-TSLN LPcaML obtains its low-dimensional local coordinates using PCA technique, at last LPcaML unifies all local coordinates to global low-dimensional coordinates while processing all neighborhoods sequentially in α-TSLN.

The idea of LPcaML algorithm is simple and intuitive. Moreover, it requires no extra constraint on underlying manifold. In fact, the geometric intuition which LPcaML is based on is an inevitable proposition of the manifold assumption. In addition by using the divide and conquer strategy, LPcaML algorithm can learn from incremental data and can process large scale data set efficiently.

The remainder of this paper is organized as follows: some notations and a simple example are given in section 2; Section 3 describes the details of LPcaML; Section 4 shows experiments results of LPcaML, as well as comparison of LPcaML with other algorithms; Conclusions and future directions are included in section 5.

2 Notations and a Simple Example

First, we give some notations and a simple example to illustrate LPcaML algorithm.

Let $X = \{x_1, x_2, \cdots, x_n\}, x_i \in R^D$ is the high-dimensional input data. Assume d is dimensionality of underlying manifold which the data lie on or close to, the goal of the LPcaML algorithm is to obtain the low-dimensional representation of the data in R^d, i.e. $Y = \{y_1, y_2, \cdots, y_n\}$, where $y_i \in R^d (i = 1, 2, \cdots, n)$.

There are three steps in LPcaML: 1) finding an α-TSLN of input data set into local neighborhoods; 2) for each neighborhood in the α-TSLN, obtaining low-dimensional local coordinates of it by using PCA on each neighborhood; 3) transforming local coordinates to a global unified coordinates by processing each neighborhoods in their appeared order in the sequence.

Fig.1 is a simple example which illustrates the basic idea and the main steps of LPcaML algorithm. The 100 data points shown in subfigure a are sampled from 3/4 unit circle at random. First, after giving the local neighborhood search algorithm such as kNN or ε-NN, LPcaML finds a traversal sequence of local neighborhoods with overlapping factor 0.4 of those 100 data points. The 0.4-TSLN is showing in subfigure b, and each ellipse is a local neighborhood. Note that in each local neighborhood, the data points approximately lie on a line, so by using classical PCA on each local neighborhood, LPcaML obtains their 1D local coordinates, which faithfully preserves the geometric and topological property of that local neighborhood. At last LPcaML transforms all local coordinates into unified global coordinates by processing each neighborhoods in their appeared order in the sequence 0.4-TSLN. In this procedure, for current being processed neighborhood LPcaML first computes the transformation function by least square fit between the global coordinates and the local coordinates of the overlapped data points, then using the transformation function transforms local coordinates of the rest data points of that neighborhood to unified global coordinates. When all the local neighborhoods have been traversed, LPcaML obtains unified global 1D coordinates of all these 100 data points showing in subfigure c.

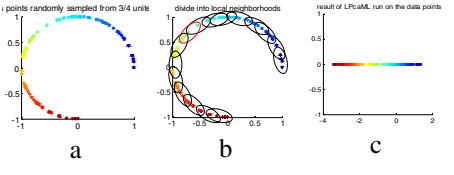


Fig. 1. Illustration of the basic idea of LPcaML algorithm

3 Detailed Implementations of LPcaML Algorithm

In this section, the details of LPcaML algorithm will be given.

3.1 α-TSLN of *X*

First LPcaML algorithm finds a traversal sequence of local neighborhood of X with overlapping factor α (α-TSLN). $\mathcal{S} =< C_1, C_2, \cdots, C_m >$ is an α-TSLN of X if it satisfies the following conditions:

1). $C_j (j = \{1, 2, \cdots, m\})$ is a neighborhood of one point in X.

2). $\bigcup_{j=1}^{m} C_j \supset X$

3). $\forall C_j (j > 1), \#(\bigcup_{i<j} C_i \cap C_j) > \alpha \# C_j$.

An extreme α-TSLN can be containing n neighborhoods, one neighborhood for each data point. But this α-TSLN has too many redundant, and many of them can be removed because all the data points in them have been covered by others. Moreover, it will increase the computational complexity. So we should minimize $\#\mathcal{S}$, this is a variation of the classical minimum set cover problem which is a NP-complete problem. But there is no need for us to find the optimal solution, so we design the following approximate algorithm, whose basic strategy is to add the neighborhood which can bring the maximal number of new points each time into the sequence. Fig.2 is the sketch of the algorithm.

Sketch of algorithm for finding α-TSLN of X

Input: high-dimensional data set X, neighborhood set $\mathcal{C}$, overlapping factor α

Output: α-TSLN of X $\mathcal{S}$.

Initialization:

$\mathcal{S} =< C_1 >$, where $\#C_1 = \max\{\#C_i, C_i \in \mathcal{C}\}$

$CurrentC = C_1$;

Iteration:

while $X \not\subset \bigcup_{C_j \in \mathcal{S}} C_j$,

do for each neighborhood C in $\mathcal{C}$

If $C \cap CurrentC \neq \varnothing$ and $\#(C \cap (X \setminus \bigcup_{C_j \in \mathcal{S}} C_j)) < (1-\alpha) \cdot \#C$

Then add to candidate set *CondS*.

select a neighborhood C from *CondS* and add it to $\mathcal{S}$, where $C = \arg\max_{C \in CondS} \#(C \cap (X \setminus \bigcup_{C_j \in \mathcal{S}} C_j))$

$CurrentC = C$

Fig. 2. Sketch of algorithm for finding an α-TSLN of X

3.2 Local Principal Component Analysis

Now assume $\mathcal{S} = \{C_1, C_2, \cdots, C_m\}$ is an α-TSLN of X. How to get low-dimensional local coordinates of data points in each neighborhoods? The basic geometric intuition is that all data points in each neighborhood reside on or close to d-dimensional linear space even if they are represented as D-dimensional vector in input space. So for each local neighborhood $C_j (j = 1, 2, \cdots, m)$, we can exploit classical PCA technique on it to obtain low-dimensional local coordinates of data points while preserving the geometric and topological property of neighborhood C_j. Suppose $C_j = (x_{j_1}, x_{j_2}, \cdots, x_{j_k})$ is the local neighborhood data matrix(we use $C_j (j = 1, 2, \cdots, m)$ as set and matrix

interchangeably), and $U_j = (u_{j,1}, u_{j,2}, \cdots, u_{j,d})$ is the first d principal components projection matrix, then the d-dimensional local coordinates of C_j are as follows:

$$Y'_j = (y'_{j_{j_1}}, y'_{j_{j_2}}, \cdots, y'_{j_{j_k}}) = U_j{}^T C_j \tag{1}$$

3.3 Local Least Square Fit

How to compute the transformation function which transforms the local coordinates Y'_j of C_j to global unified coordinates Y_j is a key issue in step 3 of LPcaML.

First LPcaML uses the first neighborhoods C_1 in the sequence $\mathcal{S}$ as base neighborhood, its unified global coordinates matrix Y_1 just is its local coordinates matrix Y'_1. And for each $C_j (j > 1)$, LPcaML first computes the transformation function. All the neighborhoods $C_i (i < j)$ have been processed before C_j, and the global coordinates of data points in $\bigcup_{i<j} C_i$ have been obtained. Without loss of generality, denote $\bigcup_{i<j} C_i \cap C_j = \{x_{j_1}, x_{j_2}, \cdots, x_{j_p}\}$ and the global low-dimensional coordinates matrix of those overlapped data points is $Y_{j_c} = (y_{j_1}, y_{j_2}, \cdots, y_{j_p})$, the local coordinates matrix of those overlapped data points in the local neighborhood C_j is denoted as $Y'_{j_c} = (y'_{j_1}, y'_{j_2}, \cdots, y'_{j_p})$, in C_j the rest of data points matrix is $Y'_{j_r} = (y'_{l_1}, y'_{l_2}, \cdots, y'_{l_q})$. Note that local coordinates Y'_{j_c} and global coordinates Y_{j_c} can be seen as different representations of the same data points in different coordinate charts of the same linear space. So the local coordinates Y'_{j_c} can be transformed to global coordinates Y_{j_c} by linear transformation. Assume the linear transformation matrix is B_j, it can be solving by minimizing the following local least square fit error between global coordinates Y_{j_c} and local coordinates Y'_{j_c} of the overlapped data points:

$$\min_{B_j} \left\| B_j \cdot (\mathbf{1} \quad Y'^{T}_{j_c})^T - Y_{j_c} \right\|_F^2 \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\mathbf{1}$ is a vector with all elements 1. This problem has only one optimized solution only if matrix $(\mathbf{1} \quad Y'^{T}_{j_c})$ is full column rank, which requires that $(\mathbf{1} \quad Y'^{T}_{j_c})$ has more than (d+1) rows. This is equivalent to that $\bigcup_{i<j} C_i \cap C_j = \{x_{j_1}, x_{j_2}, \cdots, x_{j_p}\}$ has more than (d+1) data points. So the overlapping factor can't be too small. The optimized solution can be expressed in the following formula:

$$B_j = Y_{j_c}((\mathbf{1} \quad (Y'_{j_c})^T)^+)^T \tag{3}$$

where $(\mathbf{1} \quad (Y'_{j_c})^T)^+ = ((\mathbf{1} \quad (Y'_{j_c})^T)^T (\mathbf{1} \quad (Y'_{j_c})^T))^{-1} (\mathbf{1} \quad (Y'_{j_c})^T)^T$.

After obtaining the transformation matrix, the low-dimensional global coordinates Y_{j_r} can be obtained by following expression:

$$Y_{j_r} = B_j(\mathbf{1} \ (Y'_{j_r})^T)^T \tag{4}$$

and when all local neighborhood in α-TSLN have been processed, LPcaML algorithm obtains low-dimensional global representation of all data points in *X*.

3.4 Incremental Learning Ability

Now most existing manifold learning algorithms can't learn from data that collected in sequence such as data stream. LPcaML algorithm divides the entire data set into local neighborhood and obtains low-dimensional representation by processing the local neighborhoods in sequence, so it has satisfactory performance in learning from incremental data.

Suppose the incremental data is $\triangle D$, for different type data points LPcaML has different methods to obtain low-dimensional representations.

First for the data points in regions from which the previous data are sampled, each of them can be classified into a corresponding local neighborhood in $\mathcal{S}$. For the ε-NN neighbors search algorithm, LPcaML directly classifies each this type data point to one of the neighborhoods whose ε-covering region contains it. For the kNN algorithm, note that the essential requirement for neighborhoods is that its covering region of the underlying manifold approximately lies on d-dimensional space but not the number of data points in it, so we can also classifies the data point into one of the neighborhoods whose covering region contain it. Assume that data point Δx_i is in local neighborhood C_j, LPcaML first obtains the local coordinates $\Delta y'_i$ of Δx_i using the previous results of PCA on C_j by the following formula:

$$\Delta y'_i = U_j^T \Delta x_i \tag{5}$$

and then the global coordinates Δy_i can be computed by the following formula:

$$\Delta y_i = B_j(1 \ \Delta y_i'^T)^T \tag{6}$$

Second for the data point are not in any local neighborhood of $\mathcal{S}$, we can use the extrapolation method. Denote matrix $\Delta X = (\Delta x_1, \Delta x_2, \cdots, \Delta x_{\Delta n})$ consists of all those data points, for each of them we first find its neighborhood (contains itself) in $\Delta X \cup X$ and then select a neighborhood which has no less than (*d*+1) overlapped examples with previously processed data set *X*. If we can find such a neighborhood, denote as ΔC_1, then we can find an α-TSLN $\Delta \mathcal{S}$ of ΔX with ΔC_1 as the first neighborhood using the algorithm described in Fig.2, at last LPcaML can obtain a global coordinates of this incremental data just only considering $\Delta \mathcal{S}$ as additional sequence linked to the tail of $\mathcal{S}$. But if there is no such a neighborhood, we can think that this incremental data reside on a new manifold and also we can discovery the new manifold of this incremental data using LPcaML. This method provides a new approach to research the evolution of manifold.

4 Experiments

We have conducted many experiments on synthetic data sets and real world face data sets to test our LPcaML performance. Here we only give part of the experiments results. For all data sets we fixed the overlapping factor as α=0.4.

4.1 Synthetic Data Sets

We selected three synthetic data sets including swiss roll, swiss roll with hole and twin peaks to do our experiments, and we also selected some algorithms, such as ISOMAP[8], LLE[9, 15], Laplacian eigenmaps[10], HLLE[16], LTSA[11], RML[6], to compare with LPcaML. The implementations of these algorithms are provided by the Mani Matlab Demo[17] except RML which is provided by the authors.

First we compare the outputs of each algorithm on given data set. For each data set, we chose 1000 data points at random, and our LPcaML can give good results on them. Here, we show the results of each algorithm on two data sets. In the following, we describe the comparative results in detail.

1. Swiss roll with hole (Fig.3). Again the results show that LPcaML and RML can obtain satisfactory embeddings preserving the shape of both manifold and the hole. However, the shape of the hole is distorted by ISOMAP and LLE. Moreover, the hole is magnified by ISOMAP. Although both HLLE and LTSA preserve the hole, they compress the data set into a square region again. Laplacian eigenmaps produce a line style shape.

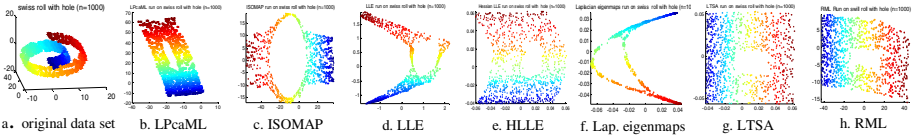


Fig. 3. Results of each algorithm run on Swiss roll with hole data set(n=1000)

2. Twin peaks (Fig.4). For this data set, all algorithm yield good result. Because curvature of underlying manifold varies significantly, a sufficient minimal neighborhood parameter is necessary to guarantee data points in each local neighborhood lie on low-dimensional linear space. But a too minimal neighborhood parameter will bring on neighborhoods that don't satisfy the imposed constraint (see the analysis of section 3). This makes manually selecting neighborhood parameter for LPcaML more difficult. So we must find a method to adaptively select neighborhood parameter based on curvature of underlying manifold. This is our future direction.

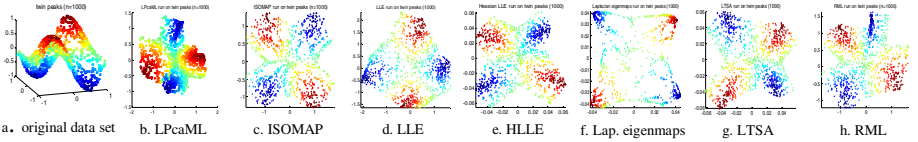


Fig. 4. Results of each algorithm run on twin peaks data set(n=1000)

Then we conducted experiments on several data sets with small number of examples (n=100-500). The goal of this type experiment is to compare the necessary approximate minimal number of examples for each algorithm to yield correct outputs. Table.1 shows the results of each algorithms. To yield correct result on a given data set, our LPcaML needs the least examples, the next is RML. And Fig.5 shows the outputs of some algorithms on swiss roll data set with 200 examples, we can see that only our LPcaML algorithm can produce a good result.

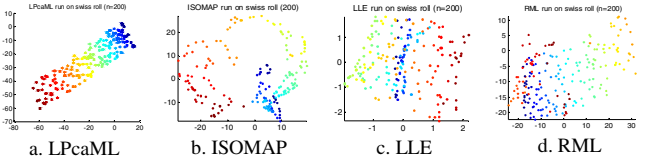


Fig. 5. The results of several algorithms run on swiss roll with 200 examples

Table 1. The necessary approximate minimal number of examples for each algorithm to yield correct results

	swiss roll	swiss roll with hole	s-curve
LPcaML	200	250	200
ISOMAP	400	400	200
LLE	400	400	400
Laplacian eigenmaps	400	400	300
LTSA	400	400	300
RML	300	300	200

The above extensive experiments show that our LPcaML algorithm can yield faithful and satisfactory results on most data sets. Moreover, most manifold learning algorithms yield good results only on dense sampled examples, while experiments demonstrate that our LPcaML can yield good results on relative less number of examples.

4.2 Real World Data Set

To evaluate our LPcaML on real world data set, we used two face data sets to test LPcaML. The two data sets are ISOMAP face data and LLE face data which are originally used by ISOMAP and LLE respectively.

The intrinsic degrees of freedom of this face data is 3, so the goal is to discover a 3-dimensional embedding. Fig.6 is the result of LPcaML on ISOMAP face data. In the subfigure a it is the projection of the 3D embedding on the XY plane while in subfigure b it is the projection on YZ plane. The representative images shown in the figure are corresponding to the red circle points. From the result, we can see that our LPcaML can output intuitive interpretable result and moreover the result represents the intrinsic degrees of freedom. For example, in subfigure a the representative facial images in the right-up region are the similar poses: faces turning to right, then in the

middle region are faces looking at front, while in the left-bottom are facial images turning to left. This convinces that the x-axis is correlated to the horizontal photograph angle. The analogous conclusion can be produced in the subfigure b.

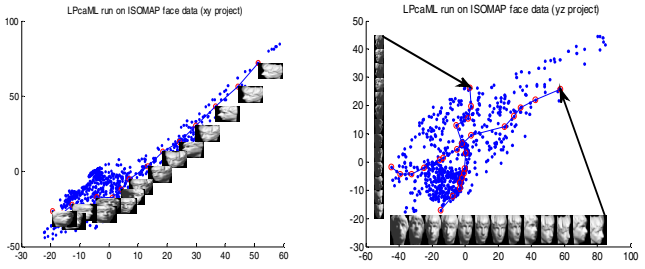


Fig. 6. The results of LPcaML on ISOMAP face data

On the LLE face data, our LPcaML outputs the similar results, due to space limitations we don't show the results here.

5 Conclusions and Future Direction

In this paper, a novel manifold learning algorithm LPcaML with incremental learning ability is proposed. The divide and conquer strategy provides LPcaML algorithms with incremental learning ability. Experiments on many synthesis data sets, as well as two real-world face data sets, demonstrate that LPcaML algorithm can learn the intrinsic structure of the data and yield regular embeddings. Special experiments are also conducted with sparsely sampled data on the synthesis data sets, results show that LPcaML algorithm can still discovery the manifold while other existing algorithms can't. Moreover, the incremental learning ability significantly extends the application areas of the LPcaML algorithm.

However, both the neighborhood parameter and the intrinsic dimensional of the manifold are manually determined before LPcaML algorithm processes. In the future we will pursue a self-adaptive parameter selecting method for LPcaML algorithm. In addition, an effective approach to traverse the local neighborhoods can also influence the quality of embeddings, so a stable and effective traverse method will be the goal for us in the future.

Acknowledgements

The authors would like to thank Tong Lin for providing RML algorithm source code. This work is supported by the National Natural Science Foundation of China(NO.60603015).

References

1. Donoho, D.: High-dimensional data analysis: the curse and blessings of dimensionality American Math Society on Math Challenges of the 21st century, Los Angeles (2000)
2. Jolliffe, I.T.: Principal Component Analysis. Springer, Heidelberg (1989)
3. Cox, T., Cox, M.: Multidimensional Scaling. Chapman and Hall, Boca Raton (1994)
4. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (2001)
5. Hastie, T., Stuetzle, W.: Principal Curves. J. Am. Statistical Assoc. 84, 502–516 (1989)
6. Lin, T., Zha, H.: Riemannian Manifold Learning. IEEE Trans. Pattern Anal. Mach. Intell. 30(5), 796–809 (2008)
7. Seung, H.S., Lee, D.D.: The manifold ways of perception. Science 290(22), 2268–2269 (2000)
8. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(22), 2319–2323 (2000)
9. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290(22), 2323–2326 (2000)
10. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. 15(6), 1373–1396 (2003)
11. Zhang, Z., Zha, H.: Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. SIAM J. Scientific Computing 26(1), 313–338 (2005)
12. Dollár, P., Rabaud, V., Belongie, S.: Non-isometric manifold learning: analysis and an algorithm. In: ICML 2007, pp. 241–248 (2007)
13. Bengio, Y., Paiement, J.F., Vincent, P., et al.: Out-of-Sample Extensions for LLE, ISOMAP, MDS, Eigenmaps, and Spectral Clustering. Advances in Neural Information Processing Systems 16, 177–184 (2003)
14. Law, M.H.C., Jain, A.K.: Incremental Nonlinear Dimensionality Reduction by Manifold Learning. IEEE Trans. Pattern Anal. Mach. Intell. 28(3), 377–391 (2006)
15. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research 4, 119–155 (2003)
16. Donoho, D.L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proc. Natl. Acad. Sci. U. S. A. 100(10), 5591–5596 (2003)
17. Wittman, T.: Mani Matlab Demo (2008), `http://www.math.umn.edu/~wittman/mani/`

Intelligent 3D Face Recognition

Chao Li

Electronic Engineering Technology Program
Florida A&M University
Tallahassee, Florida, 32307, USA
chao.li@famu.edu

Abstract. Face recognition technology has been a focus both in academia and industry for the last couple of years because of its wide potential application and its importance to meet the security needs of today's world. This paper proposes a method to tackle an important problem in 3D face recognition: the deformation of facial geometry that results from the expression changes of a subject. A framework composed of three subsystems: expression recognition system, expressional face recognition system and neutral face recognition system is proposed and implemented. This framework enables more intelligent face recognition. The recognition of faces that were neutral or exhibited one expression, i.e. smiling, was tested on a database of 30 subjects. The results proved the feasibility of this framework.

Keywords: face recognition, biometrics, 2D, 3D, range image, PCA, subspace, SVM, LDA.

1 Introduction

Face recognition is one branch of the research area known as 'biometric identification' or 'biometrics'. Examples of other biometric modalities include fingerprint, voice, iris, gait, etc. Biometrics refers to identifying an individual based on his or her distinguishing characteristics. Face recognition is a particularly compelling biometric modality because it is the one used every day by nearly everyone as the primary means for recognition of other humans. Face recognition has the advantage of being noninvasive.

Face recognition has a variety of potential applications for commercial, security, and forensic purposes. These applications include automated crowd surveillance, access control, credit card authorization, design of human computer interfaces, etc. Especially, the surveillance systems rely on the noninvasiveness of face recognition systems. Face recognition approaches can be categorized into 2D face recognition, 3D face recognition and infrared face recognition modalities based on the format of the data acquired. The infrared face recognition is commonly combined with other biometric technologies.

Most of the face recognition attempts that have been made until recently use 2D intensity images by photographic cameras as the data format for processing. This kind of research is called 2D face recognition. Varying levels of success have been

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 416 – 425, 2008.
© Springer-Verlag Berlin Heidelberg 2008

achieved in 2D face recognition research. Detailed and comprehensive surveys can be found in [1, 2]. Although 2D face recognition has achieved considerable success, certain problems still exist. Because the 2D face images used not only depend on the face of a subject, but also depend on the imaging factors, such as the environmental illumination and the orientation of the subject. These two sources of variability in the face image often make the 2D face recognition system fail. That is the reason why 3D face recognition is believed to have an advantage over 2D face recognition.

With the development of 3D imaging technology, more and more attention has been directed to 3D face recognition. In [3], Bowyer et al. provide a survey of 3D face recognition technology. Some of the techniques are derived from 2D face recognition, such as PCA used in [4, 5] to extract features from faces. Some of the techniques are unique to 3D face recognition, such as the geometry matching method in [6] and the profile matching proposed in [7, 8]. Most of the 3D face recognition systems treat the 3D face surface as a rigid surface. But actually, the face surface is deformed by different expressions of the subject. Therefore, systems that treat the face as a rigid surface are prone to fail when dealing with faces with expressions. In [9], experiments using Iterative Closest Point (ICP) and PCA methods were performed on the recognition of faces with expression. The authors found that expression changes do cause performance to deteriorate in all methods.

Therefore, the involvement of facial expression has become a big challenge in 3D face recognition systems. Up to now, only some methods address the facial expression issue in face recognition. In [10], the authors present a 3D face recognition approach based on a representation of the facial surface , invariant to isometric deformation by facial expression. In [11], both rigid registration and non-rigid deformations caused by expression were calculated. For the purpose of face matching, the non-rigid deformations from different sources are identified, which is formulated as a two-class classification problem: intra-subject deformation vs. inter-subject deformation. The deformation classification results are integrated with the matching distance of rigid registration to make the final decision. In [9], the author tried to extract the area that deforms least with different facial expressions and used this area as the feature for every subject.

In this paper, the author tackles the expression challenge in 3D face recognition from a different point of view. Because the deformation of the face surface is always associated with specific expression, an integrated expression recognition and face recognition system is proposed. This system has more intelligence in terms of using different approach to recognize faces with different expressions. In section 2, a model of the relationship between expression and face recognition is introduced. Based on this model, the framework of integrated expression recognition and face recognition is proposed. Section 3 explains the acquisition of the experimental data used and preprocessing performed. Section 4 outlines our approach to 3D face expression recognition. Section 5 explains the process used for 3D face recognition. Section 6 describes the experiments and the results obtained. Section 7 is presents our discussion and conclusion.

2 Expression Recognition and Face Recognition

From the psychological point of view, it is still not known whether facial expression recognition information aids the recognition of faces by human beings. One of the experiments that support the existence of the connection between facial expression recognition and face recognition was reported in [12]. The authors found that people are slower in identifying happy and angry faces than they are in identifying faces with neutral expression.

The proposed framework involves an initial assessment of the expression of an unknown face, and uses that assessment to facilitate the recognition of faces. The incoming 3D range image is processed by an expression recognition system to find the most appropriate expression label for it. The expression labels include the six prototypical expressions of the faces, which are happiness, sadness, anger, fear, surprise and disgust[13], plus the neutral expression. Therefore, the output of the expression recognition system will be one of the seven expressions. According to different expressions, a matching face recognition system is proposed. If the output is neutral expression, then the incoming 3D range image is directly passed to the neutral expression face recognition system, which uses the features of the probe image to directly match those of the gallery images, which are all neutral instances of all enrolled subjects, to get the closest match. If the output is other than neutral expression, then for each of the six prototypical expressions, a separate face recognition subsystem should be used. The system will find the right face through modeling the variations of the face features between the neutral face and the face with expression. Since the recognition through modeling is a more complex process than the direct matching for the neutral face, this framework would support the experimental findings, which showed that people are slower in identifying happy and angry faces than they are in identifying faces with neutral expression. Figure 1 shows a simplified version of this framework. This simplified diagram only deals with the happy (smiling) expression, which is the most commonly displayed by people publicly.

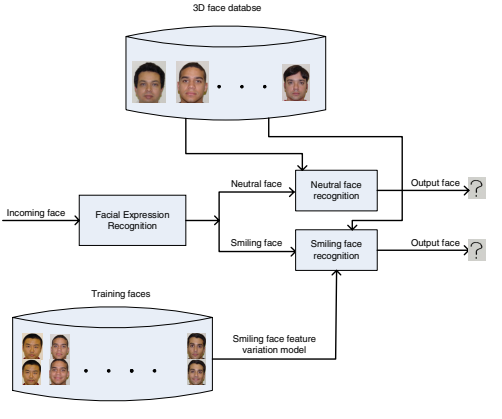


Fig. 1. Simplified framework of 3D face expression

3 Data Acquisition and Preprocessing

To test the idea proposed in this model, a database, which includes 30 subjects, was built. In this database, we test the different processing of the most common expression, i.e., smiling versus neutral. Each subject participated in two sessions of the data acquisition process, which took place in two different days. In each session, two 3D scans were acquired. One was a neutral expression; the other was a happy (smiling) expression. The 3D scanner used was a Fastscan 3D scanner from Polhemus Inc. [14]. The resulting database contains 60 3D neutral scans and 60 3D smiling scans of 30 subjects. The left image in Figure 2 shows an example of the 3D scans obtained using this scanner.

In 3D face recognition, registration is a key pre-processing step. In our experiment, a method based on the symmetric property of the face is used to register the face image[15]. Because in the scanning process there were some unwanted holes in the face surface, (especially in the area covered by dark hair, such as the eye brows), a cubic spline interpolation method was used to patch the holes. An example of the resulting 3D range images is shown in Fig 2(right image).

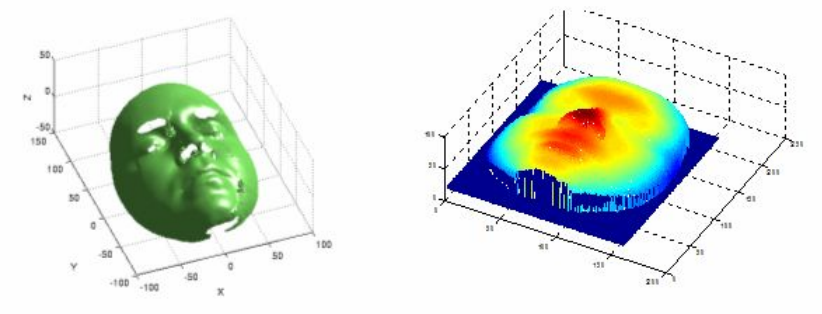


Fig. 2. 3D surface (left) and mesh plot of the converted range image (right)

4 Expression Recognition

The expression of the face is a basic mode of nonverbal communication among people. The facial expressions convey information about emotion, mood and ideas. In [13], Ekman and Friesen proposed six primary emotions. Each possesses a distinctive content together with a unique facial expression. These prototypical emotional displays are also referred to as basic emotions. They seem to be universal across human ethnicities and cultures. These six emotions are happiness, sadness, fear, disgust, surprise and anger. Together with the neutral expression, these expressions also form the seven basic prototypical facial expressions.

Automatic facial expression recognition has gained more and more attention recently. It has various potential applications in improved intelligence in human computer interfaces, image compression and synthetic face animation. “Face expression recognition deals with the classification of facial motion and facial feature deformation into abstract classes that are purely based on visual information.” [16].

As in face recognition, most contemporary facial expression systems use two-dimensional images or videos as data format. Logically, the same 2D shortcomings will hamper 2D expression recognition, (i.e., 2D formats are dependent on the pose of the subjects and on the illumination of the environment). In this respect this paper fills the gap by proposing a facial expression system that uses three dimensional images or range images, which are invariant with respect to illumination and subject orientation.

In our experiment, we aim to recognize social smiles, which were posed by each subject. Smiling is the easiest of all expressions to find in photographs and is readily produced by people on demand. It is generated by contraction of the zygomatic major muscle. The zygomatic major originates in the cheek bone (zygomatic arch) and inserts in muscles near the corner of the mouth. This muscle lifts the corner of the mouth obliquely upwards and laterally producing a characteristic "smiling expression". So, the most distinctive features associated with the smile are the bulge of the cheek muscle and the uplift of the corner of the mouth as we can see from the following photos. The line on the face generated by the smiling expression is called the nasal labial fold (smile line).

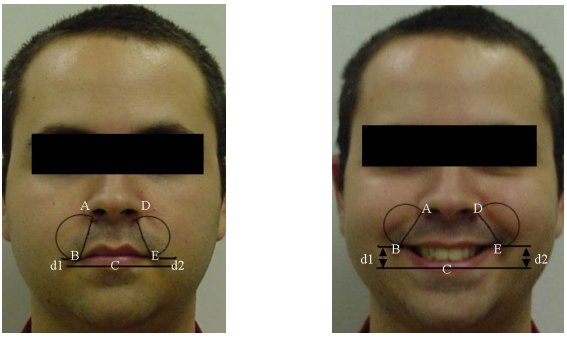


Fig. 3. Illustration of features of smiling face

The following steps are followed to extract the features for the smiling expression:

1. An algorithm is developed to obtain the coordinates of five characteristic points A, B, C, D and E in the face range image as shown in Figure 3. A and D are at the extreme points of the base of the nose. B and E are the points defined by the corners of the mouth. C is in the middle of the lower lip.
2. The first feature is the width of the mouth BE normalized by the length of AD. Obviously, while smiling the mouth becomes wider. The first feature is represented by *mw*.
3. The second feature is the depth of the mouth (The difference between the Z coordinates of points BC and EC) normalized by the height of the nose to capture the fact that the smiling expression pulls back the mouth. The second feature is represented by *md*.
4. The third feature is the uplift of the corner of the mouth, compared with the middle of the lower lip d1 and d2, as shown in the figure, normalized by the

difference of the Y coordinates of points AB and DE, respectively and represented by *lc*.

5. The fourth feature is the angle of AB and DE with the central vertical profile, represented by *ag*.
6. The last two features are extracted from the semicircular areas shown, which are defined by using AB and DE as diameters. The histograms of the range (Z coordinates) of all the points within these two semicircles are calculated.

The following figure shows the histograms for the smiling face and the neutral face of the above subject.

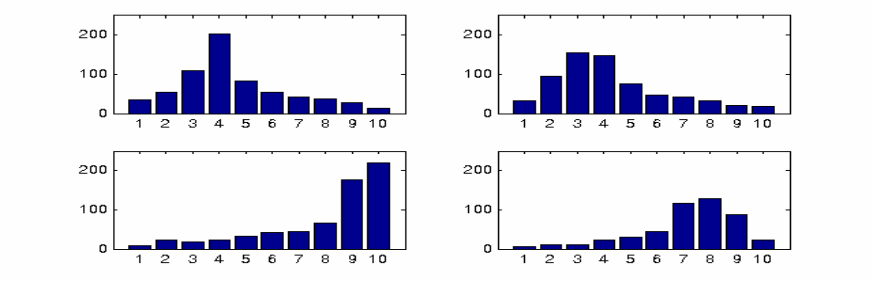


Fig. 4. Histogram of range of cheeks for neutral (top row), and smiling (bottom row) face

The two figures in the first row are the histograms of the range values for the left cheek and right cheek of the neutral face image; the two figures in the second row are the histograms of the range values for the left cheek and right cheek of the smiling face image.

From the above figures, we can see that the range histograms of the neutral and smiling expressions are different. The smiling face tends to have large values at the high end of the histogram because the bulge of the cheek muscle. On the other hand, a neutral face has large values at the low end of the histogram distribution. Therefore two features can be drawn from the histogram:

One is called the 'histogram ratio', represented by *hr*, the other is called the 'histogram maximum', represented by *hm*.

$$hr = \frac{h6 + h7 + h8 + h9 + h10}{h1 + h2 + h3 + h4 + h5} \tag{1}$$

$$hm = i \quad ; \qquad i = \arg\{\max(h(i)\} \tag{2}$$

In summary, six features, i.e. *mw*, *md*, *lc*, *ag, hr* and *hm* are extracted from each face for the purpose of expression recognition.

After the features have been extracted, this becomes a general classification problem. Two pattern classification methods are applied to recognize the expression of the incoming faces. The first method used is a linear discriminant classifier, which seeks the best set of features to separate the classes. The other method used is a support vector machine. Support vector machine is a relatively new technology for classification. It relies on preprocessing the data to represent patterns in a high dimension,

typically much higher than the original feature space. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two categories can always be separated by a hyperplane. [17] For our work, Libsvm [18] was used to implement a suitable support vector machine.

5 3D Face Recognition

5.1 Neutral Face Recognition

In earlier research work, It was found that the central vertical profile and the contour are both discriminant features for every person[15]. Therefore, for neutral face recognition, the same method as in [19] is used: the results of central vertical profile matching and contour matching are combined. The combination of the two classifiers improves the overall performance significantly. The final similarity score for the probe image is the product of ranks for each of the two classifiers (based on the central vertical profile and contour). The image with the smallest score in the gallery will be chosen as the matching face for the probe image.

5.2 Smiling Face Recognition

For the recognition of smiling faces we have adopted the probabilistic subspace method proposed by B. Moghaddam et al. [20, 21]. It is an unsupervised technique for visual learning, which is based on density estimation in high dimensional spaces using an eigen decomposition. The probability density is used to formulate a maximum-likelihood estimation framework for visual search, target detection and automatic object recognition. Using the probabilistic subspace method, a multi-class-classification problem can be converted into a binary classification problem.

In the experiment for smiling face recognition, because of the limited number of subjects (30), the central vertical profile and the contour are not used directly as vectors in a high dimensional subspace. Instead, they are down sampled to a dimension of 17 to be used. The dimension of difference in feature space is set to be 10, which contain approximately 97% of the total variance. The dimension of difference from feature space is 7.

The results of central vertical profile matching and contour matching are combined. Here also the combination of the two classifiers improves the performance. The final similarity score for the probe image is the product of ranks for each of the two classifiers. The image with the smallest score in the gallery will be chosen as the matching face for the probe image.

6 Experiments and Results

One gallery and three probe databases are formed for the evaluation of our methods in three experiments. The gallery database has 30 neutral faces, one for each subject, recorded in the first data acquisition session. Three probe sets are formed as follows:

Probe set 1: 30 neutral faces acquired in the second session.
Probe set 2: 30 smiling faces acquired in the second session.
Probe set 3: 60 faces, constituted by probe1 and probe2.

Experiment 1: Testing the expression recognition module
The leave one out cross validation method is used to test the expression recognition classifier. Every time, the faces collected from 29 subjects in both data acquisition sessions are used to train the classifier and the four faces of the remaining subject collected in both sessions are used to test the classifier. The results shown below are the average of the 30 recognition rates. Two classifiers are used. One is the linear discriminant classifier; the other is a support vector machine classifier.

Table 1. Expression recognition result

Method	LDA	SVM
Expression recognition rate	90.8%	92.5%

Experiment 2: Testing the neutral and smiling recognition modules separately
In the first two sub experiments, probe faces are directly fed to the neutral face recognition module. In the third sub experiment, the leave-one-out cross validation is used to verify the performance of the smiling face recognition module. In each cycle, 29 subjects' faces from both acquisition sessions are used for the training and the remaining subject's smiling face from the session is used as testing face.

2.1 Neutral face recognition: probe set 1. (Neutral face recognition module used.)
2.2 Neutral face recognition: probe set 2. (Neutral face recognition module used.)
2.3 Smiling face recognition: probe set 2. (Smiling face recognition module used.)

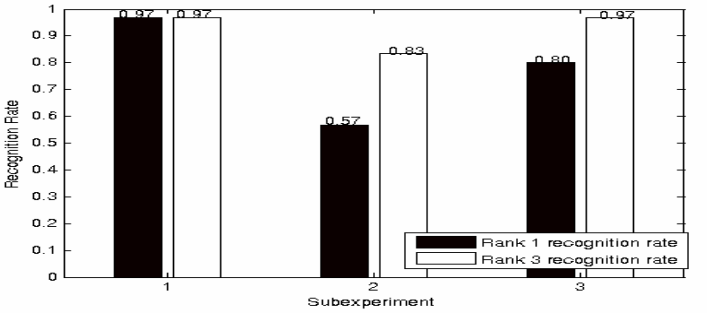


Fig. 5. Results of Experiment 2

From Figure 5, it can be seen that when the incoming faces are all neutral, the algorithm which treats all the faces as neutral achieves a very high recognition rate. On the other hand, if the incoming faces are smiling faces, then the neutral face recognition algorithm does not perform well, only 57% rank one recognition rate is obtained.

In contrast, when the smiling face recognition algorithm is used to deal with smiling faces, the recognition rate can be as high as 80%.

Experiment 3: Testing a practical scenario
These experiments emulate a realistic situation in which a mixture of neutral and smiling faces (probe set 3) must be recognized. Sub experiment 1 investigates the performance obtained if the expression recognition front end is bypassed, and the recognition of all the probe faces is attempted with the neutral face recognition module alone. The last two sub experiments implement the full framework shown in Figure 1. (Faces are first sorted according to expression and then routed to the appropriated recognition module.) In 3.2 the expression recognition is performed with the linear discrimant classifier, while in 3.3 it is implemented through the support vector machine approach.

3.1 Neutral face recognition module used alone: probe set 3 is used
3.2 Integrated expression and face recognition: probe set 3 is used. (Linear discriminate classifier for expression recognition.)
3.3 Integrated expression and face recognition: probe set 3 is used. (Support vector machine for expression recognition.)

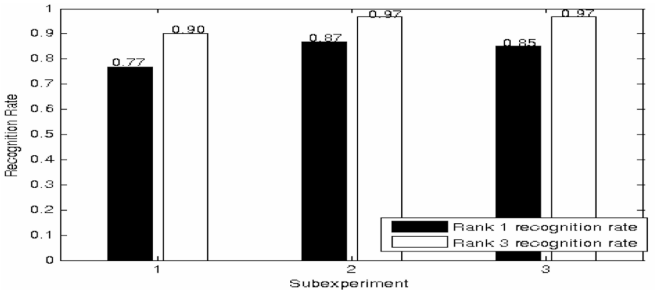


Fig. 6. Results of Experiment 3

It can been seen in Figure 6 that if the incoming faces include both neutral faces and smiling faces, the recognition rate can be improved about 10 percent by using the integrated framework proposed here.

7 Discussion and Conclusion

In this paper we have presented an intelligent 3D face recognition system by acknowledging the fact that the face of a subject is a deformable surface that undergoes significant changes when the subject displays common expressions. Our main proposition is that, instead of ignoring the possibility of significant facial changes due to expressions, 3D face recognition systems should account for the potential presence of an expression in their probe images. The experiment results proved the feasibility of this method. This method has the advantage of being computationally efficient. Furthermore, this method also yields the information of the expression found in the faces.

References

1. Chellappa, R., Wilson, C., Sirohey, S.: Human and machine recognition of faces: a survey. Proceedings of the IEEE 83(5), 705–740 (1995)
2. Zhao, W., Chellappa, R., Rosenfeld, A.: Face recognition: a literature survey. ACM Computing Survey 35, 399–458 (2003)
3. Bowyer, K., Chang, K., Flynn, P.: A Survey of Approaches to 3D and Multi-Modal 3D+2D Face Recognition. In: IEEE International Conference on Pattern Recognition, pp. 358–361 (2004)
4. Chang, K., Bowyer, K., Flynn, P.: Multimodal 2D and 3D biometrics for face recognition. In: IEEE International Workshop on Analysis and Modeling of Faces and Gestures, pp. 187–194 (2003)
5. Hesher, C., Srivastava, A., Erlebacher, G.: A novel technique for face recognition using range images. In: Seventh International Symposium on Signal Processing and Its Application (2003)
6. Gordon, G.: Face recognition based on depth maps and surface curvature. Geometric Methods in Computer Vision, SPIE, 1–12 (July 1991)
7. Cartoux, J., LaPreste, J., Richetin, M.: Face authentication or recognition by profile extraction from range images. In: Proceedings of the Workshop on Interp. of 3D Scenes, pp.194–199 (1989)
8. Nagamine, T., Uemura, T., Masuda, I.: 3D facial image analysis for human identification. In: Internaitonal Conference on Pattern Recognition (ICPR 1992), pp. 324–327 (1992)
9. Chang, K., Bowyer, K., Flynn, P.: Effects of facial expression in 3D face recognition. SPIE: Biometric Technology for Human Identification II (2005)
10. Bronstein, A., Bronsein, M., Kimmel, R.: Expression-invariant 3D face recognition. In: Proc. Audio & Video-based Biometric Person Authentication (AVBPA), pp. 62–69 (2003)
11. Lu, X., Jain, A.: Deformation analysis for 3D face matching. In: 7th IEEE Workshop on Applications of Computer Vision WACV 2005 (2005)
12. Etcoff, N., Magee, J.: Categorical perception of facial expressions. Cognition 44, 227–240 (1992)
13. Ekman, P., Friesen, W.: Constants across cultures in the face and emotion. Journal of Personality and Social Psychology 17(2), 124–129 (1971)
14. http://www.polhemus.com.
15. Li, C., Barreto, A.: Profile-based 3D face registration and recognition. In: Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 478–488. Springer, Heidelberg (2005)
16. Fasel, B., Luettin, J.: Automatic facial expression analysis: a survey. Pattern Recognition 36, 259–275 (2003)
17. Duda, R., Hart, P., Stork, D.: Pattern classification, 2nd edn (2001)
18. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001)
19. Li, C., Barreto, A., Zhai, J., Chin, C.: Exploring face recognition using 3D profiles and contours. In: IEEE SoutheastCon 2005, pp. 576–579 (2005)
20. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object detection. In: International Conferrence of Computer Vision (ICCV 1995), pp. 786–793 (1995)
21. Moghaddam, B., Pentlend, A.: Probabilistic visual learning for object representation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(7), 696–710 (1997)

Language Feature Mining for Music Emotion Classification via Supervised Learning from Lyrics

Hui He[1,2], Jianming Jin[2], Yuhong Xiong[2], Bo Chen[1], Wu Sun[3], and Ling Zhao[3]

[1] School of Information and Communication Engineering, Bejing University of Posts and Telecommunications, 100876 Beijing, P.R. China
{hh1012,chb615}@gmail.com
[2] HP Labs China, 100022 Beijing, P.R. China
{jian-ming.jin,yuhong.xiong}@hp.com
[3] YY Music Group
{xiaoge.sun,ling.zhao}@yy.com

Abstract. In recent years, efficient and intelligent music information retrieval became very important. One essential aspect of this field is music emotion classification by earning from lyrics. This problem is different from traditional text categorization in that more linguistic or semantic information is required for better emotion analysis. Therefore, we focus on how to extract useful and meaningful language features in this paper. We investigate three kinds of pre-processing methods and a series of language grams having different n-order under the well-known n-gram language model framework to extract more semantic features. Then, we employ three supervised learning methods (Naïve Bayes, maximum entropy classification, and support vector machines) to examine the classification performance. Experimental results show that feature extraction methods improve music emotion classification accuracies. Maximum entropy classification with unigram+bigram+trigram gets best accuracy and it is suitable for music emotion classification.

Keywords: emotion classification, n-grams, Naïve Bayes, Maximum Entropy, Support Vector Machine.

1 Introduction

The rapid growth of the Internet and the advancements of Internet technologies have made it possible for music listeners to have access to a large amount of online music data, including music sound signals, lyrics, biographies, and so on. This raises the question of whether computer programs can enrich the experience of music listeners by enabling the listeners to have access to such a large volume of online music data. Multimedia conferences, e.g. the International Conference on Music Information Retrieval (ISMIR) and Web Delivery of Music (WEBELMUSIC), have a focus on the development of computational techniques for analyzing, summarizing, indexing, and classifying music data.

Traditionally musical information has been retrieved and/or classified based on standard reference information, such as the name of the composer and the title of the work

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 426–435, 2008.
© Springer-Verlag Berlin Heidelberg 2008

etc. But these are far from satisfactory. Huron points out that since the preeminent functions of music are social and psychological, the most useful characterization would be based on four types of information: the style, emotion, genre, and similarity [1].

The emotional component of music has attracted interest in the Music Information Retrieval (MIR) community, and experiments have been conducted to classify music by mood [2, 3, 4, 5]. Some investigate audio signals and some use lyrics to explore music emotion classification. In this paper, we choose to learn from lyrics for two main reasons. First, it is easy to get lyrics from Internet. Some websites provide free services to search, download and add lyrics. Second, it is much easier to process lyrics than audio signals since the preprocessing of audio signals is more complicated and time-consuming than lyrics. Thus it is more suitable for computing in mobile or MP3, MP4, etc.

However, previous work [6] suggests that lyrics are difficult for natural language processing. First, unlike other text data such as news, emotion words in lyrics have a very small number of occurrences. Instead, a large portion will be devoted to the background. This makes the term absolute frequency information not so useful, or even misleading. Second, because lyrics are free-style, approaches based on word positions also face difficulty. Third, lyrics downloaded from Internet may contain spelling mistakes, because most of lyrics are added by listeners. Therefore, in this paper, we adopt the popular n-gram language models and part of speech (POS) to mine kinds of language features. And then we employ three supervised learning methods to examine the classification performance.

The rest of this paper is organized as follow: In section 2, a briefly review of previous work on music emotion classification is presented. In section 3, how the n-gram based language features are extracted and weighted are described. Supervised learning methods are introduced in section 4. Section 5 presents our experiments and results analysis. Conclusions are derived in section 6.

2 Related Work

Music emotion classification has attracted interest in recent years. In related work, some researchers investigate audio signals and some use lyrics to explore music emotion classification. Thus, in this section, we will take a brief review of some related works.

2.1 Learning from Audio Signals

Relations between musical sounds and their impact on the emotion of the listeners have been studied for decades. The celebrated paper of Hevner [7] studied this relation through experiments in which the listeners are asked to write adjectives that came to their minds as the most descriptive of the music played. The experiments confirmed a hypothesis that music inherently carries emotional meaning.

Li [2, 3, 4] introduced Daubechies Wavelet Coefficient Histograms (DWCH) for music feature extraction to learn from audio signals and conducted a comparative study of sound features and classification algorithms on music emotion classification.

By combining DWCH with timbral features (MFCC and FFT), SVM achieved better performance.

Lu [8] presented a hierarchical framework to automate the task of mood detection from acoustic music data. Three feature sets, including intensity, timbre, and rhythm were extracted to represent the characteristics of a music clip. Preliminary evaluations indicated that the proposed algorithm produced satisfactory results.

2.2 Learning from Lyrics

Most previous work learning from lyrics is on stylometric analysis. Li [4] studied the problem of identifying groups of artists by combining acoustic-based features and lyrics-based features and using a semi-supervised classification algorithm.

Work of music emotion classification via lyrics is rare. Lyrics is a special kind of text, so learning from lyrics can be taken as sentiment classification via texts. Similar work has been reported in [9] and improved work in [10]. In these two papers, the language feature extraction methods and machine learning techniques they used are of great help of our work.

3 Language Feature Extraction

Document representation is an important aspect in traditional topic-based categorization. In normal documents, there are always enough items to form term-vectors, while in lyrics, fewer items are there, especially for emotion words. Thus, we need to apply language model to capture more language features to express these lyrics.

3.1 Language Features

(1) Basic unigram. In topic-based categorization, the standard word-vector or so called bag-of-words is the basic and mostly used representation of document. For example, in a text snippet T consist of n words can be represented as a word set $\{w_1, w_2, \cdots, w_n\}$.

While being put into the n-gram framework, it can be viewed as a basic unigram form, which is a context free document representation model. In such model, each item is assumed to be independent of any other ones, thus the dependencies among items are absolutely ignored.

(2) Conventional n-gram. In a conventional n-gram ($n > 1$) model, the last $n-1$ items are viewed as the history information of the current item. It's obvious that a high order n-gram model can capture some short-distance dependencies by combining n sequential individual terms into a compounded feature, e.g., in a bigram model, sentence T will have a feature set $\{w_1w_2, w_2w_3, \cdots, w_{n-1}w_n\}$.

In this music emotion classification task, it can be noted that in addition to many words with obvious emotion tendencies, there are also a lot of tendentious phrases consist of two or three words. While a unigram model can make use of the individual words, n-gram models are good at taking advantage of those phrases or compounds, which are much more meaningful in emotion analysis.

(3) Part of speech. It is found in previous work that only a few words, whose part-of-speech are adjective, verb, noun, and adverb, can present text sentiment. So, one way of extracting features is to filter lyrics only retain words mentioned above.

3.2 Feature Selection and Weighting

As mentioned above, the feature set may consist of all possible language-grams. The original amount of these features will be too huge for an efficient document analysis system. On the other hand, a great percent of these language-grams are nothing but sequential words, which have neither linguistic structure nor semantic meaning and should be cleared out of our language model. Additionally, different language-grams are of different linguistic functions and semantic meanings, so they could not be viewed equally. Therefore, there are two procedures need to be perform on the original language-grams, one is feature selection, and the other is feature weighting.

Our emphasis was put on finding out what kinds of language features are useful. Thus, we mainly applied a simple feature elimination method that is to abandon those feature appearing fewer times than a threshold.

For feature weighting, we try three feature weighting methods: Boolean value, absolute term frequency and term frequency-inverse document frequency (TFIDF) weighting method. The following equation (1) is TFIDF weighting used in this paper:

$$w(f_k, l_i) = \frac{tfidf(f_k, l_i)}{\sqrt{\sum_{f_k} (tfidf(f_k, l_i))^2}} \tag{1}$$

$$tfidf(f_k, l_i) = \sqrt{tf(f_k, l_i)} \times \log\left(\frac{|D|}{df(f_k)}\right) \tag{2}$$

In equation (2), D is the corpus including all lyrics. $tf(f_k, l_i)$ is frequency of feature f_k occurs in lyrics l_i. $df(f_k)$ is lyrics frequency of feature f_k occurs in whole corpus. Equation (1) is normalization of equation (2).

4 Supervised Learning Methods

Our aim in this work is to examine whether it is helpful in music emotion classification by feature extraction from lyrics. We employ three standard algorithms: Naïve Bayes classification, maximum entropy classification, and support vector machines. The main ideas behind these three algorithms are quite different, but each has been shown to be effective in previous text categorization.

We use the following standard bag-of-features framework. Let $\{f_1, f_2, \cdots, f_m\}$ be a predefined set of m features that can appear in a lyrics. A feature can be a word or bigram etc. Let $n_i(l)$ be the number of times f_i occurs in lyrics l. Then, each lyrics l is represented by the lyric vector $\vec{l} = weighting(n_1(l), n_2(l), \cdots, n_m(l))$, $weighting()$ is weighting methods mentioned in section 3.2.

4.1 Naïve Bayes

The Naïve Bayes classifier (NB for short) which is based on a simple application of Bayes rule is a simple but effective machine learning algorithm. It performs very well while being applied to text classification [11, 12]. Apply it to music emotion classification with lyrics, it can be presented as:

$$P(c \mid l) = \frac{P(c) \times P(l \mid c)}{P(l)} \tag{3}$$

where c denotes emotion category and l denotes lyrics. $P(c)$ is the prior distribution of a category. NB can be constructed by seeking the optimal category which maximizes the posterior $P(c \mid d)$.

Assume all of the attribute values are independent to the given category label. In addition $P(d)$ is a constant for every emotion category c, we can get:

$$c^* \propto \arg\max_{c \in C} \left\{ P(c) \times \prod_{j=1}^{m} P(f_j \mid c) \right\} \tag{4}$$

where a lyrics l is represented by a vector of m features which are treated as features appearing in the lyrics l, $\vec{l} = weighting(n_1(l), n_2(l), \cdots, n_m(l))$. $P(f_j \mid c)$ stands for the probability that the feature f_j occurs in a category c in training data, and Laplace smoothing method is usually chosen to estimate it to overcome the zero-frequency problem.

4.2 Maximum Entropy

Maximum entropy classification (MaxEnt, or ME, for short) is an alternative technique which has proven effective in a number of natural language processing applications [13]. Its estimate of $P(c \mid l)$ takes the following exponential form:

$$P_{ME}(c \mid l) = \frac{1}{Z(l)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(l, c) \right) \tag{5}$$

where $Z(l)$ is a normalization function. $F_{i,c}$ is a feature/class function for feature f_i and class c, defined as follow:

$$F_{i,c}(d, c') = \begin{cases} 1, & n_i(d) > 0 \quad and \quad c' = c \\ 0, & otherwise \end{cases} \tag{6}$$

Importantly, unlike Naïve Bayes, maximum entropy classification makes no assumptions about the relationships between features, and so might potentially perform better when conditional independence assumptions are not met.

The $\lambda_{i,c}$'s are feature-weight parameters; inspection of the definition of P_{ME} shows that a large $\lambda_{i,c}$ means that f_i is considered a strong indicator for class c. The

parameter values are not set so as to maximize the entropy of the induced distribution (hence the classifier's name) subject to constraint that the expected values of the feature/class functions with respect to the model are equal to their expected values with respect to the training data: the underlying philosophy is that we should choose the model making the fewest assumptions about the data while still remaining consistent with it, which makes intuitive sense.

4.3 Support Vector Machine

The support vector machine (SVM) is a powful supervised learning algorithm developed by Vapnik[14]. It has been successfully applied to text classification and performed very well. In its simplest form, the goal of a linear SVM is to find the hyperplane which can split different category examples by maximizing the distance between the nearest examples to the hyper-plane. Using a kernel function, the nonlinear SVM maps the input variables into a high dimensional space, and linear SVM can be applied in that space. In the binary classification, the corresponding decision function is :

$$F_{i,c}(d,c') = \begin{cases} 1, & n_i(d) > 0 \quad and \quad c' = c \\ 0, & otherwise \end{cases} \tag{7}$$

where K is a kernel function. Linear, polynomial kernel, Gaussian RBF kernel and sigmoid kernel are the typical kernel functions usually used in SVM.

5 Experiments

5.1 Dataset

In this paper, the music emotion classification was based on the lyrics of Chinese pop music, which is provided by YY Music Group. This data set consists of 1,903 songs. 803 songs are labeled with emotion of love, and 1100 songs are labeled as lovelorn songs. We measured the performance of different methods mentioned above using 5-fold cross validations.

5.2 Experiment Scheme

To prepare the lyrics, we automatically segmented these Chinese lyrics. We tried our methods on three kinds of segmented data sets. First data set was just segmented lyrics and no stop lists were used. Second was segmented lyrics having been deleted stop words[1], which were extensively used. Third data set was segmented lyrics with POS, and was filtered by POS and only retained words with POS of adjective, verb, noun, and adverb.

We attempted to model the potentially important contextual features. For this study, we focused on features based on unigrams, bigrams and trigrams. Since a great percent of these language-grams were nothing but sequential words, which had neither linguistic structure nor semantic meaning and would be cleared out of our language model, we ignored features that occurred 5 or fewer times in dataset.

[1] http://bbs.langtech.org.cn/

The implementation we used for maximum entropy classification is modified on the basis of SS Maxent toolkit[2]. We used Chih-Chung Chang's LIBSVM[3] package for training and testing, with all parameters set to their default values. In this paper, we employ linear kernel as kernel function.

5.3 Experiment Results and Analysis

As described in section 3.2, first, we tested the three weighting methods as Boolean, absolute term frequency, and TFIDF. To make a comparision, these methods have been applied on three kinds of datasets and with their unigram, unigram+bigram and unigram+bigram+trigram language feature sets respectively.

Experiment results are shown in Table1~Table3 with three different preprocessing datasets mentioned in experiment scheme. In these tables, the mark "ABS" stands for the absolute term frequency weighting, "BOOL" for Boolean form, and "TFIDF" for term frequency-inverse document frequency weighting method. Results are measured by accuracy.

(1) Different supervised learning algorithms. As a whole, the supervised learning algorithms clearly surpass the random-choice baseline of 50%. Maximum entropy classification and SVM perform much better than Naïve Bayes. In three different preprocessing datasets, accuracies of maximum entropy classification and SVM with same features and weighting methods are nearly the same.

(2) Different preprocessing. As can be seen from Table 1 and Table 2, a little improvement has been made by deleting stop words with same conditions (including same features, weighting methods and supervised learning algorithms). It seems that some stop words are noisy data in music emotion classification. Thus, if a better result is required, an elaborate stop word list should be selected.

Compared with Table 1 and Table 3, accuracies of lyrics with POS filtered processing decline a little with cases of Naïve Bayes and maximum entropy classification. ME with features of unigrams+bigrams+trigrams obtains best accuracy of all, which

Table 1. Average 5-fold cross-validation accuracies (Lyrics segmented only)

Features	Weighting	NB	ME	SVM
Unigrams	ABS	64.37%	86.71%	**86.76%**
Uni+bigrams	ABS	66.74%	87.86%	**88.07%**
Uni+bi+trigrams	ABS	68.94%	87.39%	**87.91%**
Unigrams	BOOL	64.37%	86.71%	**87.49%**
Uni+bigrams	BOOL	66.74%	87.86%	**88.07%**
Uni+bi+trigrams	BOOL	68.94%	87.39%	**88.49%**
Unigrams	TFIDF	58.85%	**89.28%**	88.56%
Uni+bigrams	TFIDF	54.70%	**89.28%**	88.81%
Uni+bi+trigrams	TFIDF	53.86%	**89.65%***	89.28%

[2] Offered by Tsujii laboratory, at University of Tokyo, on http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/maxent/

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 2. Average 5-fold cross-validation accuracies (Lyrics segmented and deleting stopwords)

Features	Weighting	NB	ME	SVM
Unigrams	ABS	68.94%	**87.97%**	86.60%
Uni+bigrams	ABS	70.84%	**88.65%**	87.70%
Uni+bi+trigrams	ABS	70.94%	**88.49%**	88.28%
Unigrams	BOOL	68.94%	**87.97%**	87.81%
Uni+bigrams	BOOL	70.84%	88.65%	**88.75%**
Uni+bi+trigrams	BOOL	70.94%	88.86%	**89.23%**
Unigrams	TFIDF	61.32%	88.49%	**89.02%**
Uni+bigrams	TFIDF	55.33%	89.18%	**89.34%**
Uni+bi+trigrams	TFIDF	54.91%	88.70%	**89.21%**

Table 3. Average 5-fold cross-validation accuracies (Lyrics segmented with POS filtered)

Features	Weighting	NB	ME	SVM
Unigrams	ABS	64.30%	**86.49%**	85.76%
Uni+bigrams	ABS	65.72%	**87.38%**	86.71%
Uni+bi+trigrams	ABS	66.14%	**87.65%**	86.23%
Unigrams	BOOL	64.30%	**86.49%**	86.39%
Uni+bigrams	BOOL	65.72%	87.38%	**88.39%**
Uni+bi+trigrams	BOOL	66.14%	87.65%	**87.81%**
Unigrams	TFIDF	59.99%	88.70%	**89.20%**
Uni+bigrams	TFIDF	55.42%	89.12%	**89.43%**
Uni+bi+trigrams	TFIDF	54.84%	89.17%	**89.33%**

shows that high order of grams captures more semantic features, so best results are achieved. However, SVM with POS filtered increases, especially SVM with features of unigrams+bigrams and TFIDF weighting get the highest accuracy in Table 3. We suppose that the retained words with POS of adjective, verb, noun, and adverb represent main emotion expressed by songs.

(3) Different weighting methods. In general, the performances of TFIDF are obviously better than Boolean weighting and absolute term frequency in the three tables. This may due to that the TFIDF weighting is a discount weighting method between the Boolean weighting and absolute term frequency weighting. It makes the feature space more smoothing between high frequency terms and low frequency terms while still remaining differences among features.

Interestingly, accuracies of Naïve Bayes with TFIDF weighting drop 10%. It is conflicted with traditional topic categorization. We speculate that this perhaps due to topic being conveyed mostly by particular content words that tend to be repeated, but this remains to be verified in music emotion classification with Naïve Bayes.

(4) Different n-gram features. We also studied the use of bigrams and trigrams to capture more context in general. It can be seen from these three tables that all accuracies with same algorithm and weighting methods are improved by adding high order

grams except Naïve Bayes with TFIDF weighting. We suspect that this is because TFIDF weighting is not suitable for Naïve Bayes.

6 Conclusions and Future Work

In this paper, we focus on how to extract useful and meaningful language features. We investigate three kinds of preprocessing methods and a series of language grams having different n-order under the well-known n-gram language model framework. Then we employ three prevail supervised learning methods to examine the classification performance. Experiment results show that feature extraction methods improve music emotion classification accuracies. ME with unigram+bigram+trigram get best accuracy and it is suitable for music emotion classification.

The work reported here is still worth in-depth studying. The n-gram based feature representation is useful, but feature sets contain noise and useless grams. In this paper, we just ignore them by feature frequency. Some supervised feature selection and extraction methods could be adopted to improve accuracy and efficiency.

Acknowledgements

This research is partially supported by the National High-tech Development Plan under Grant No.2007AA01Z417, and the 111 Project under Grant No. B08004. The authors would like to thank YY music group for provision of music emotion corpus, who has spent great efforts in intelligence music service.

References

1. Huron, D.: Perceptual and Cognitive Applications in Music Information Retrieval. In: Proc. Int. Symp. Music Information Retrieval (2000)
2. Li, T., Ogihara, M.: Detecting Emotion in Music. In: Proc. Fifth Int. Symp. Music Information Retrieval (ISMIR 2003), pp. 239–240 (2003)
3. Li, T., Ogihara, M.: Content-based Music Similarity Search and Emotion Detection. In: Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, pp. 705–708 (2004)
4. Li, T., Ogihara, M.: Toward Intelligent Music Information Retrieval. IEEE Transactions on Multimedia 8(3), 564–574 (2006)
5. Skowronek, J., McKinney, M., van de Par, S.: A Demonstrator for Automatic Music Mood Estimation. In: Proc. 8th Int. Symp. Music Information Retrieval (ISMIR 2007) (2007)
6. Scott, S., Matwin, S.: Text Classification Using WordNet Hypernyms. In: COLING-ACL 1998 Workshop, pp. 38–44 (1998)
7. Hevner, K.: Experimental Studies of the Elements of Expression in Music. Amer. J. Psychol. 48, 246–268 (1936)
8. Lu, L., Liu, D., Zhang, H.: Automatic Mood Detection and Tracking of Music Audio Signals. IEEE transactions on audio, speech, and language processing 14(1), 5–18 (2006)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification Using Machine Learning Techniques. In: Proceedings of the Conference on Empircal Methods in Natural Language Processing, Philadelphia, US, pp. 79–86 (2002)

10. Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cus. In: Proceedings of 42nd Meeting of the Association for Computational Linguistics, Barcelona, ES, pp. 271–278 (2004)
11. Mccallum, A., Nigam, K.: A Comparison of Event Models for Naïve Bayes Text Classification. In: Proceedings of AAAI 1998 Workshop on Learning for Text Categorization, pp. 41–48 (1998)
12. Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R.: Tackling the Poor Assumption of Naïve Bayes Text Classifiers. In: Proceedings of the 20th International Conference on Machine Learning (ICML 2003) (2003)
13. Berger, A.L., Della Pietra, S.A., Della Pietra, V.J.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics 22(1), 39–71 (1996)
14. Vapnik, V.: Principles of Risk Minimization for Learning Theory. In: Lippman, D.S., Moody, J.E., Touretzky, D.S. (eds.) Advances in Neural Information Processing Systems, pp. 831–838. Morgan Kaufmann, San Francisco (1992)

An Immune Multi-agent System for Network Intrusion Detection

Dian Gang Wang[1,2], Tao Li[1], Sun Jun Liu[1], Gang Liang[1], and Kui Zhao[1]

[1] School of Computer Science, Sichuan University, Chengdu 610065, China
wangdg@vip.sina.com
[2] Sichuan Electric Vocational and Technical College, Chengdu 610072, China

Abstract. Inspired by the immune theory and multi-agent systems, an immune multi-agent system for network intrusion detection is established. The concept of immune agent is introduced. And its logical structure and running mechanism are established. This model implements the multi-layer and distributed mechanism for network intrusion detection. The experimental results show that the new model not only reduces the False-Negative rate and False-Positive rate effectively but also has the feature to adapt to continuous changing network environments.

1 Introduction

With the rapid development of information technology and Internet, many network intrusion technologies have continuously changed, which result in more and more serious damages. The network security has been the focus of the public attention. But the present technologies, e.g. accessing controlling, certification authorization, firewall, etc. can only provide a passive defense, which do little to the dynamically changed Internet. Intrusion Detection System, referred to as IDS, has been an indispensable component of the network information security defense system. And it is the focus of the network security researches.

However, these intrusion detection technologies[1][2][3], including statistics analysis, feature analysis, specialist system, data mining, etc., all have some limitations[1]. They lack self-adaptation and can only detect the known attacks other than new attack models. The system also lacks robustness, which means that even the error in a local place can affect the whole works. And each part of the system is isolated with each other, so that there is not an effective communication.

An immune multi-agent system for network intrusion detection is proposed here, referred to as IMASNID, which makes use of self-adaptation, diversity, memory ability in artificial algorithm[4-7], and combines the robustness and distribution in multi-agent system's architecture[3][8-9]. The concept of immune agent (IA) and its logical structure model are presented, and the IA running mechanism is established. And the concept of bacterin [10] in biological system is introduced. The experiment results show that this model is a good solution to the network security.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 436–445, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Related Work

2.1 Artificial Immune System

The Human Immune System (HIS) protects the body against damage from an extremely large number of harmful bacteria and viruses, termed pathogens. It does this largely without prior knowledge of the structure of these pathogens. This property, along with the distributed, self-organized and lightweight nature of the mechanisms by which it achieves this protection, has in recent years made it the focus of increased interest within the computer science and network security communities. Seen from such a perspective, the HIS can be viewed as a form of anomaly detector with very low False- Positive and False-Negative rates. An increasing amount of work is being carried out attempting to understand and extract the key mechanisms through which the HIS is able to achieve its detection and protection capabilities. A number of artificial immune systems(AIS) have been built for a wide range of applications including document classification, fraud detection, network- and host-based intrusion detection, optimization and industry control fields. These AIS have acquired some successful achievements in specific domain.[11]

2.2 Summary of Multi-agent System

Agent [12] is an entity that has the ability of consciousness, solving problems and communication. With apperceiving local environment, it can adjust by itself according to the internal reasoning mechanism. With the cooperation and coordination during the isolated agents in a Multi-agent system [9], the problems in the complex environment can be solved and it has the features of distribution, robustness.

3 The Theory of IMASNID

As an intelligent-independent-autonomous entity, immunocytes have the similarities with agent in essence. And a biological immune system can be seen as a distributed multi-agent system. So we use immune and multi-agent theory to detect intrusion.

3.1 Immune Agent

An immune system is an autonomous one which owns features of distribution and multi-agent. AIS has the abilities of self-learning and memory. And immunocytes are distributed and independent. In this case, a new concept of agent that is immune agent, referred to as IA, is introduced here. Besides the common features inherited from the general agent, IA [4-5] has the desirable features of evolution, identification diversity, memory, tolerance, etc.. [13-16]

3.2 The Architecture of IMASNID

The proposed model constructs a multi-layer intelligent network security system, using multi-agent technologies and AIS. Its architecture is shown in Fig.1, where IA

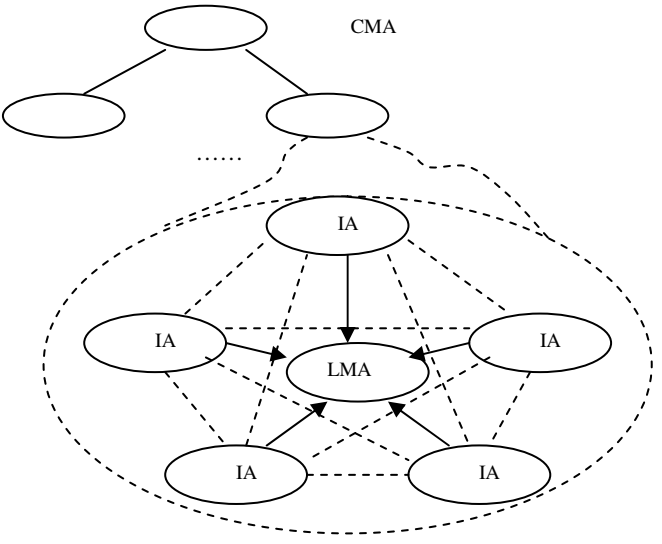


Fig. 1. The Architecture of IMASNID

is the security state of computers surveilled. Local Monitor Agent (LMA) analyzes the state of the local area network(LAN), while Central Monitor Agent(CMA) surveils the whole network.

IA is the kernel of this model, distributing in every host node, identifying the intrusion affairs, and founding unknown attacks through learning and memory. It also quantitatively evaluates the risk state facing by the node and sends the state to its own network segment's Local Monitor Agent (LMA). Meanwhile, it would send the bacterin of the new attack to each node in the same network segment, which improves the ability of intrusion detection.

LMA takes charge of surveilling the network segment, mixing data for each IA's information, evaluating the risk status of its own network segment, and sending it to Central Monitor Agent (CMA). And LMA of the intrusion segment would send bacterin to some other segments without intrusion to warn them. CMA takes charge of human computer interaction. It receives the intrusion status of network segment sent by LMA and shows the secure status through client interface. According to this, the administrator could take some response measures to control and protect the whole network timely.

3.3 Intrusion Detection Mechanism of the IA

Fig. 2 shows the architecture and work flow of IA, which consists of self antigens, immature antibodies, mature antibodies and memory antibodies, etc. The work flow includes two kinds of procedure: the procedure of antigen detected and the procedure of immune antibodies generated.

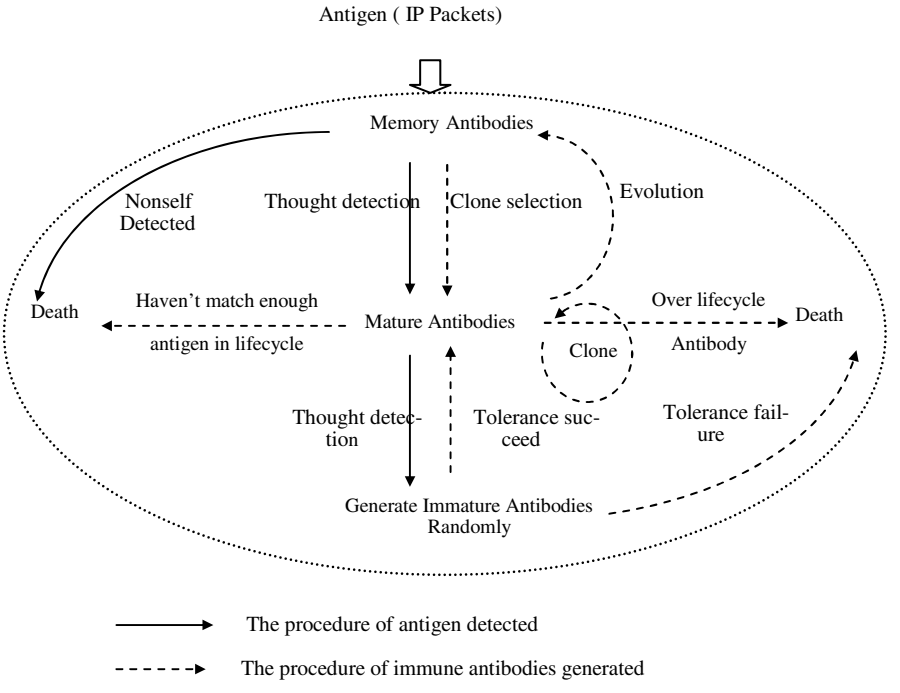


Fig. 2. The architecture and work flow of IA

3.3.1 The Definition of Immune Elements

Definition 1. Antigens in our approach are binary strings extracted from the IP packets, including IP address, port number, protocol type, etc. Set $U = \{0,1\}^l \ (l > 0)$, and $Ag \subset U$. And the set U can be divided into self which indicates the normal network behaviors and nonself which indicates the abnormal network affairs, where:

$$Self \cup Nonself = Ag, Self \cap Nonself = \Phi \tag{1}$$

Definition 2. Antibodies are binary strings having the similar features and length with antigens, which is defined as $B \subset U$:

$$B = \{Ab \mid Ab =< s, age, count, ag >, s, ag \in U \wedge age, count \in N\} \tag{2}$$

Where *s* is the antibody binary strings whose length is *l*, age is the cell age, count is the antigen number matched by antibody, ag is the antigen detected by the antibody, N is the set of natural number. There are three types of antibodies: mature, memory and immature ones.

Definition 3. The computation of affinity is shown in formula (3) use affinity to judge the match between antibody and antigens:

$$f_{match}(x, y) = \begin{cases} 1 & \exists i, j, j - i \geq r, 0 < i < j \leq l, x_i = y_i, \\ & x_{i+1} = y_{i+1}, ..., x_j = y_j \\ 0 & otherwise \end{cases} \tag{3}$$

where $x, y \in \{0,1\}^l$ are, respectively, an antibody and an antigen and l is the length of string,

3.3.2 The Variation Process of Immature Antibodies

Set I is the number of immature antibodies in I_{Ab}, the dynamic variation equation is:

$$I(t + \Delta t) = I(t) + I_{new} \cdot \Delta t - (\frac{\partial I_{mature}}{\partial x_{mature}} \cdot \Delta t + \frac{\partial I_{dead}}{\partial x_{dead}} \cdot \Delta t) \tag{4}$$

It shows the change of I_{Ab} can be divided into two processes: inflow and outflow. The inflow one is the process that new immature antibodies adds to the set of I_{Ab} : $I = I_{new} \cdot \Delta t$, I_{new} indicates the immature antibody's producing velocity in each time. The outflow one is the process that immature antibodies are eliminated from I_{Ab}, which contains two directions: Some would evolve into mature antibodies after self-tolerance, and $\frac{\partial I_{mature}}{\partial x_{mature}} \cdot \Delta t$ indicates the reduced immature ones' number in I_{Ab} after a successful tolerance; And the immature antibodies in I_{Ab} suffer a failed self-tolerance are also can be eliminated, whose number is defined as $\frac{\partial I_{dead}}{\partial x_{dead}} \cdot \Delta t$.

In order to prevent antibodies from matching to self, the new immature antibodies can match to antibodies only through the process of self-tolerance, where 1 means passing the self-tolerance and 0 means not, $ab \in I_{Ab}$:

$$f_{tolerate}(ab) = \begin{cases} 0 & iff\ \exists y \in Self \wedge f_{match}(ab.s, y) = 1 \\ 1 & otherwise \end{cases} \tag{5}$$

3.3.3 The Variation Process of Mature Antibodies

Set T is the number of mature antibodies in T_{Ab}, the dynamic variation equation is :

$$T(t + \Delta t) = T(t) + (\frac{\partial T_{tolerate}}{\partial x_{tolerate}} \cdot \Delta t + \frac{\partial T_{clone}}{\partial x_{clone}} \cdot \Delta t) - (\frac{\partial T_{active}}{\partial x_{active}} + \frac{\partial T_{dead}}{\partial x_{dead}}) \cdot \Delta t \tag{6}$$

It shows the change of T can divided into two processes: inflow and outflow. The inflow one is the process that mature antibodies adds to the set of T_{Ab}, which has two ways: Some are evolved from immature antibodies with self-tolerance, and the corresponsive increased number of mature ones in T_{Ab} is $\frac{\partial T_{tolerate}}{\partial x_{tolerate}} \cdot \Delta t$, where $\frac{\partial T_{tolerate}}{\partial x_{tolerate}} \cdot \Delta t = \frac{\partial I_{mature}}{\partial x_{mature}} \cdot \Delta t$. The others are evolved from memory ones after clone selection,

and the corresponsive increased number of mature ones in T_{Ab} is $\frac{\partial T_{clone}}{\partial x_{clone}} \cdot \Delta t$ .The outflow one is the process that mature antibodies are eliminated from T_{Ab} , which contains two directions: Some would be activated into memory ones, the corresponsive reduced number of mature antibodies in T_{Ab} is $\frac{\partial T_{active}}{\partial x_{active}} \cdot \Delta t$; The other would go to death if the activation fails, the corresponsive reduced number of mature antibodies in T_{Ab} is $\frac{\partial T_{dead}}{\partial x_{dead}} \cdot \Delta t$.

The set of mature antibodies that are activated into memory ones is shown in formula (8), and the set of mature ones that suffer failing activation is shown in formula (9), where β is the activated threshold, and λ is the lifecycle:

$$T_{active} := \{x \mid x \in T_{Ab} \wedge x.count \geq \beta \wedge x.age \leq \lambda\} \tag{7}$$

$$T_{dead} := \{x \mid x \in T_{Ab} \wedge x.count < \beta \wedge x.age > \lambda\} \tag{8}$$

3.3.4 The Variation Process of Memory Antibodies

Set M is the number of memory antibodies in M_{Ab} , the dynamic variation equation is:

$$M(t+\Delta t) = M(t) + \frac{\partial M_{active}}{\partial x_{active}} \cdot \Delta t + \frac{\partial M_{bacterin}}{\partial x_{bacterin}} \cdot \Delta t \tag{9}$$

Because of the infinite lifecycle in memory antibodies, the change of M_{Ab} only has the inflow process without outflow process which means the death of memory ones. The inflow has two ways: Some are transformed by the activated mature antibodies T_{active} , where $\frac{\partial M_{active}}{\partial x_{active}} \cdot \Delta t = \frac{\partial T_{active}}{\partial x_{active}} \cdot \Delta t$; Others are obtained from the antibody bacterin sent by other IA.

Once the memory one match to the antigen, it will be activated and clone itself, and has a rapid response to the antigens that have been identified before. This is the same to the second response in BIS. Clone selection would give the priority to the antibodies owning high affinity.

3.3.5 Immune Surveillance

The detection of IA on network is based on the hosts, which makes use of mature and memory cells to detect the antigens, and filters the mature ones which do well in identifying antigens to evolve into memory ones. These would make them distinguish the non-self antigens from others more quickly and more effectively. The detailed steps are as follow:

(1) Antigen extraction: Get the IP packets and extract the information including IP address, port number and protocols, etc. Construct a binary string as the antigen to add to the set of timely, whose length is *l*.

(2) Memory antibody detections: Use memory antibody to detect Ag, eliminate detected non-self. If memory antibody detects the self, it will be eliminated from M_{Ab} .

(3) Mature antibody detects antigens: Use mature antibody set T_{Ab} to detect antigens set Ag. Eliminate detected non-self from Ag. If the mature antibody detect enough antigens, it will be activated and evolve into memory one. And if the mature antibody aren't activated or detect self in its lifecycle, it will go to death.

(4) Self update: The left antigens after the above detection will be added to the self set in order to maintain updating self set. It will endure a self-tolerance process with immature set I_{Ab} to keep the dynamic circulation of antibodies' evolution.

3.4 Communication Mechanism of IMASNID

The concept of bacterin is introduced in to the active defense model. And the message mechanism makes the communication during the agent come true. The structure of the messages is defined as:

$$Message :=< Sender >< \mathrm{Re}ceiver >< SendTime >< ValidTime >< Content > \quad (10)$$

Where sender and receiver can be a transverse communication during the agents in the same level and also can be the longitudinal communication during the agents in the different levels. Receivers judge whether to receive this message depending on SendTime and ValidTime. Content contains information about new antibodies.

Distributing bacterins strengthens the contact during the agents. Sharing the effective antibodies in each IA improves the response ability of the nodes. And switching intrusion information during local monitor agents guarantees the early warning mechanism. Then the capability of resist attack is highly improved.

4 Simulations and Experiment Results

The experiment has been carried out in the Laboratory of Computer Network and Information Security at Sichuan University. A total of 40 computers in a network are under surveillance. According to the model we propose in this paper, an antigen is defined as a 96 bits length binary string composed of the source/destination IP address, port number, protocol type, IP flags, IP overall packet length, TCP/ UDP/ICMP fields, and etc. And we can use TP(True-Positive) and FP(False-Positive) to evaluate the performance of the model.

Table 1. Parameter values used for experiments

Parameters	Values
Tolerance Period	80
Life span of Mature Antibodies	50
The Size of Initial Self-Set	50
The Number of New Generated Immature Antibodies	500
The Number of Immature Antibodies Generated in the Process of Mutation	150
Generation	20000
Activation Threshold of Mature Antibodies	5

There are several important parameters in the model such as tolerance period , life span of mature antibodies, and the size of initial self-set and etc. In order to evaluate every parameter's impact on model's performance, we have carried out many group experiments. The parameters that are used in the experiments are summarized in table 1, and these values of the parameters can get relative satisfied results (FP=0.048, TP=0.962) in our experiments, which have been verified by thousands of tests in our laboratory.

In order to verify the adaptability, and test the efficiency of our model, experiments were undertaken with regards to the LISYS [17].

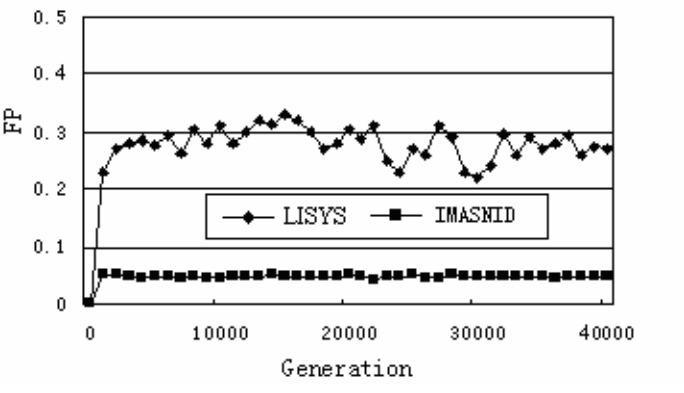


Fig. 3. FP rates for LISYS and IMASNID

As the Fig.3 shows that the FP rates of IMASNID is much lower than that of LISYS and the FP rates are not fluctuant with changing of the time. So we can safely conclude that the dynamic model is more adaptable to real network environment than LISYS.

From the following Fig.4, we can see that the TP rates of LISYS are much lower than that of IMASNID. This is mainly because that IMASNID can detect known

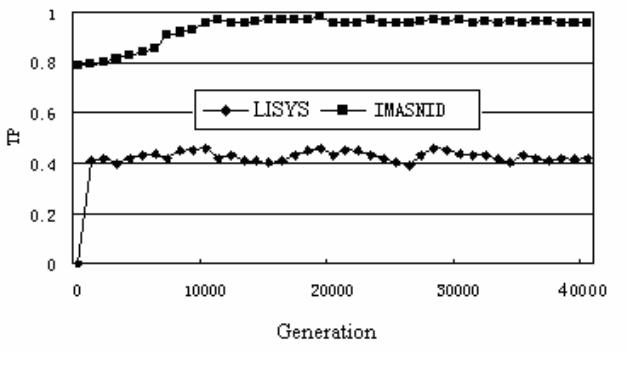


Fig. 4. TP rates for LISYS and IMASNID

intrusion rapidly and reliable. The second reason is that the elements of self set and antibodies set can vary with the changing of the network environment. So the TP rates of our model are higher than that of LISYS.

5 Conclusion

Inspired by the biological immune system, an immune multi-agent system for network intrusion is proposed. The logical structure of immune agent and its running mechanism are established, and a multi-layer and distributed architecture is achieved. In order to adapt to dynamic changing network environment, the elements of the self set and detector set can vary according to the changes of the network environment. The practical experimental results also prove that this model can efficiently reduce both the False-Positive error rates and False-Negative error rates, and enhance the ability of self-adaptation and diversity for the network intrusion detection systems. It is a good solution for the network security.

Acknowledgement

This work is supported by 863 High Tech Project of China under Grant No. 2006AA01Z435, the National Natural Science Foundation of China under Grant No.60373110, No.60573130 and 60502011.

References

[1] Bai, Y., Kobayashi, H.: Intrusion Detection Systems: technology and development. IEEE Advanced Information Networking and Applications, 710–715 (2003)
[2] Pilz, A., Swoboda, J.: Network management information models. International Journal of Electronics and Communications 58, 165–171 (2004)
[3] Dong, Y., Quian, J., Shi, M.: A cooperative intrusion detection system based on autonomous agents. In: IEEE CCECE 2003, vol. 2, pp. 861–863 (2003)
[4] Heseleer, D., Forrest, S.: An immunological approach to change detection: algorithm, analysis and implication. In: Proc. of IEEE Symposium on Research in Security and Privacy, Oakland, pp. 110–119 (1996)
[5] Kim, J., Bentley, P.: The Artificial Immune Model for Network Intrusion Detection. In: 7th European Congress on Intelligent Techniques and Soft Computing (1999)
[6] Harmer, P.K., Lamont, G.B.: An Agent Based Architecture for a Computer Virus Immune System. In: Proc. of the Genetic and Evolutionary Computation Conference, Orlando, Florida, USA (1999)
[7] Esponda, F., Forrest, S., Helman, P.: A formal frame work for positive and negative detection schemes. IEEE Transactions On Systems Man and Cybernetics Part B-Cybernetics 34(1), 357–373 (2004)
[8] Hegazy, I.M., Faheem, H.M.: Evaluating how well agent-based IDS perform. IEEE Potentials 24(2), 27–30 (2005)
[9] Ballet, P., Rodin, V.: Immune Mechanisms to Regulate Multi-Agents Systems. In: GECCO 2000, Las Vegas, Nevada, USA (2000)

[10] Jerne, N.K.: Towards a Network Theory of the Immune System. Annnual Immunology 125C, 373–389 (1974)
[11] Li, T.: Computer Immunology. Publication House of Electric Industry, Beijing (2004)
[12] Zhongzhi, S.: Intelligent agent and their application. Science Press, Beijing (2000)
[13] Hofmeyr, A., Forrest, S.: Architecture for an Artificial Immune System. Evolutionary Computation 7(1) (2000)
[14] Perelson, A.S.: Immunology for physicists. Review of Modern Physics 69(4) (1997)
[15] Dasgupta, D.: An Artificial Immune System as a Multi-Agent Decision Support System. In: Proc. of the IEEE International Conference on SMC, San Diego (1998)
[16] Li, T.: A New Model for Dynamic Intrusion Detection. LNCS, pp. 72–84 (2005)
[17] Esponda, F., Forrest, S., Helman, P.: Positive and Negative Detection. IEEE Transactions on Systems, Man and Cybernetics (2002)

A Novel Biology-Inspired Virus Detection Model with RVNS

Renchao Qin[1,2], Tao Li[1], and Yu Zhang[1]

[1] Department of Computer Science, Sichuan University, Chengdu 610065, China
[2] School of Computer Science, Southwest University of Science & Technology, Mianyang 621002, China
myxkd2000@sina.com

Abstract. A virus detection model based on artificial immune principles with real-valued negative selection (RVNS) is proposed. Feature vectors of program code are mapped into high dimension real-valued space. The architecture of this model, the formal definitions of self, nonself, antigen, antibody, and gene library are given. Then the process of self-tolerance of variable-sized detectors in real-valued space is discussed in detail. Variable-sized detectors provide concise representation of the detectors derived from benign files, which reduces the size of detectors set and advances the detection rate. The experimental results show that the model can detect obfuscated and firstly unknown virus more effectively than traditional model.

Keywords: Artificial immune principles, Virus detection, variable-seized detectors, RVNS, Antigen presentation.

1 Introduction

Since the appearance of the first computer virus in 1986, a large number of new viruses have appeared every year. With the development of Internet, the generation and spreading speed of new computer viruses are getting much faster than before. However, anti-virus software is the primary mechanism to prevent computers from the damage of virus. Such mechanism relies on the update of virus signature and scan engine to detect a new virus. Since a new virus pattern can be only extracted after the virus breakout, and small modification of the same virus can escape from the matching of the original virus pattern. For example, simple program entry point modifications consisting of two extra jump instructions effectively defeated most scanners[1]. Much attention has been paid to unknown virus detection.

The problems found in computer security systems are quite similar to the ones encountered in Biological Immune Systems (BIS). BIS has successfully solved the problem of unknown virus detection [2]. In 1994, Forrest presented a method of computer virus detection based on the negative selection algorithm[3], which is the first time to use immune mechanism for virus detecting and has greatly promoted the research of computer virus immune system (CVIS). After that many CVIS have been presented, most of them are mainly derived from ARTIS[4-7]. These CVIS in negative

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 446–453, 2008.
© Springer-Verlag Berlin Heidelberg 2008

selection used the problem in binary representation; however, many applications are natural to be described in real-valued space. Furthermore, these problems can hardly be processed properly using negative selection algorithm in binary representation[8]. In this paper, we propose a model for detection of virus based on artificial immune system with real-valued negative selection.

The rest of the paper is organized as follows. In Section 2 we introduce the concept of relevant to N-gram theory. In Section 3, we present our model based on Immune theory and real-valued negative selection. Section 4 shows our simulations and experimental results. Section 5 is the conclusions.

2 Basic N-Gram Theory

In this section we will discuss our proposal for generating gene lib. We assume that we have a set of executable files that are known to be benign codes. Our aim is to acquire a set of features of N-gram. Before extracting features of each file, we first introduce N-gram analysis. An N-gram[9] is a subsequence of N consecutive tokens in a stream of tokens. N-gram analysis has been applied in many tasks, and is well understood and efficient to implement. Figure.1 displays an example of a 2-byte window sliding right one byte at a time to generate each 2-gram. Each 2-gram is displayed in the highlighted "window".

90 [AC E4] 94 4C C9 CC E8 08 28 23 79 24 8F 24 28

90 AC [E4 94] 4C C9 CC E8 08 28 23 79 24 8F 24 28

90 AC E4 [94 4C] C9 CC E8 08 28 23 79 24 8F 24 28

Fig. 1. Sliding window (window size=2 Byte)

We use N-gram to generate gene lib set. Define gene lib set of length l as follows: $Agl = \{ag \,|\, ag \in Benign, |ag| = l, l \in N\}$, where ag is a N-gram with length of l ,it derived from benign file set, l is a natural number. If we vary l from 1 to N, we get N sets of Agl of different length. We use these sets as gene lib; it can be defined as equation (1).

$$Ag = \{Ag1 \cup Ag2 \cup, \cdots, \cup AgN\} \tag{1}$$

3 Proposed Model

Figure.2 illustrates the architecture of the proposed model. Antigens (Ag) are a vector, having the program characteristics in a computer system. This model serves to classify an input set (Ag) into self (Benign) and nonself (Virus) by mature and memory detectors[10].

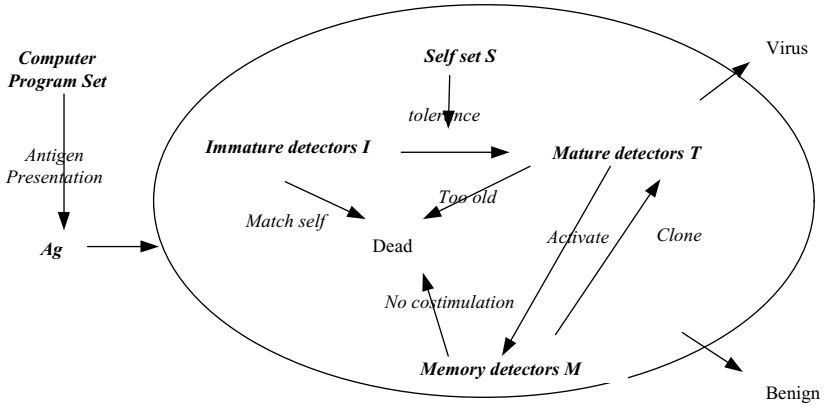


Fig. 2. Architecture of proposed model

Let $\Omega = \bigcup_{i=1}^{\infty} x^i$ be the state space of antigen, where x is a feature of files. Let self set B be the benign code set, where $B \subseteq \Omega$; and virus code set M be the code of virus, where $M \subseteq \Omega$, such that $B \cap M = \phi$, $B \cup M = Ag$. Let antigen vector be $X = (x_1, x_2, ..., x_n)$, where n is a natural number, x_i is a feature which is extracted from program file, $x_i \in [0,1]$. The task of a virus detection system is to classify an input pattern $x \in Ag$ as either B or M . This detection methodology can generate two types of errors: false-positive error and false-negative error. A false-positive error occurs when a member of B set is incorrectly classified as malicious. Conversely, a false-negative error is the classification of a member of M set as benign.

3.1 Antigen Presentation

We extract each N-gram at a given length l of the detect file, and check it in corresponding Agl . If it exists in Agl , the count increases by 1, after all of the N-gram are been processed, the ratio of N-gram of length l existing in Agl is obtain by count number divide by total l -gram size. This ratio x_i will be used as a feature of vector X , it can be defined as equation (2).

$$x_i = count / (|e| - l) \tag{2}$$

The procedure is given as follows:

```
Procedure  Antigen Presentation
Begin
count=0;
While j<|e|−l  do   /* e is detect file, |e| is it's length*/
Begin
If s∈ Agl  then    /*s is N-gram extract from current j*/
```

```
count++;
End
j++;
End
x_i =count / (|e|−l);
End.
```

3.2 Definition of Immune Cells

Self set is defined as

$$S = \{< c_s, r_s > | c \in X, r_s \in R\} \tag{3}$$

Where c_s is a vector which formed by antigen presentation process from benign codes. $X = (x_1, x_2, ..., x_n)$ is a vector, $x_i \in [0,1]$, r_s is self radius, R is a real number. Given detector set reads

$$D = \{< c_d, r_d, age, count > | c_d \in X, r_d \in R, age, count \in Z^+, age \le max_age\} \tag{4}$$

Where c_d is an antigen, r_d is the detector radius, R is a real number, age is the detector age, $count$ is the detector affinity, and max_age is the upper limit of the detector age. D is divided into immature, mature and memory detectors.

Immature detectors are newly generated ones given by

$$I = \{c_i | c_i \in D, age < \lambda, count = 0)\} \tag{5}$$

Where λ is tolerance period. Mature detectors are the ones that are tolerant to self set *S* but not activated by antigens, and given by

$$T = \{c_t | c_t \in D, \lambda \le age < max_age, count < \delta, f_{tolerance}(c_t, S) = 1\} \tag{6}$$

Where the lifecycle of mature detector is from λ to max_age, δ is the activation threshold, $f_{tolerance}$ denotes that antigen is tolerant to self (See section 3.3 Self tolerance). Memory detectors evolve from mature ones that accumulate enough affinity in their lifecycle, and given by

$$M = \{c_m | c_m \in D, age = max_age\} \tag{7}$$

3.3 Self Tolerance

To prevent immune cells from recognizing the benign code as virus in the virus detection, immature immune cell need to pass self-tolerance. Self-tolerance is carried out by negative selection algorithm.

A negative selection algorithm basically consists of two phases. First, the detector set is generated in the generation phase. Then, the new sample is examined using the detector set during the detection phase. Let us first describe the real-valued negative detection algorithm using Variable-sized detectors[11], where candidate detectors are generated randomly. Those that match any self samples (training data) using Euclidean

distance matching rule are eliminated. The generation phase finishes when a preset number of detectors are obtained. The generation phase of this algorithm is shown as follows:

```
V-Detector-Set(S,Tmax, r_s )
S:set of self samples
  Tmax:maximum number of detector
  r_s :self radius
  D = ∅
  Repeat
  t = 0
  T = 0
  r = ∞
  x = random sample from [0,1]^n
  Repeat for every d_i in D = {d_i.i = 1,2,...}
       d_d = Euclidean distance between d_i and x
       If d_d ≥ r(d_i) then /*where r(d_i) is the radius of d_i */
          Repeat for every s_i in S
          d = Euclidean distance between s_i and x
          If d − r_s ≤ r then r = d − r_s
       If r > r_s then D = D ∪ {< x, r >} /* < x, r > is a detector*/
  Until |D| = T_max
  Return D
```

3.4 Evolution of Self

$$S(t) = \begin{cases} S_{first}, & t = 0 \\ f_{s_\lim}((S(t-1) \cup S_{del}(t)) \cup S_{new}(t), & t > 0 \end{cases} \tag{8}$$

Where $S(t), S(t-1) \subset Self$ are, respectively, indicating the self-set at time t and t - 1 , S_{first} is the initial self set. $f_{s_\lim}$ is a function used to limit the number of self set: if the number of self set is larger than a given value max_s_size ,the least recently used self antigen is selected and discarded, and this procedure continues until the size of self set equal to max_s_size . $S_{del}(t)$ are the mutated self antigens discarded at time t ,which includes three parts: 1) the unloaded soft ware; 2) the elements recognized by new memory detectors; 3)the elements infected by viruses. $S_{new}(t)$ are the new self antigens (e.g., loading new software) added into self set at time t .

The new immature detectors, which are generated at random[12, 13], have to experience a self tolerance period: the detector will be eliminated if it matches any self antigens. The immature detectors that survived in self tolerance period will evolve into mature ones; there the mature detectors have a fixed lifecycle: the detectors will be

eliminated if they do not accumulate enough affinity in their lifecycle; they will be activated if they get enough affinity, i.e., viruses are found. However, the activated detectors will be eliminated if they do not receive co-stimulation, i.e., false positive error; there the detected antigens are self elements. Meanwhile, the activated detectors will evolve into memory ones with the help of co-stimulation, there the detected antigens are sure nonself elements. The memory detectors have an infinite lifecycle, and will be activated as soon as they match an antigen.

When a detector (e.g., a memory detector, or a mature one) detects a virus, it will also clone itself and create a lot of similar detectors to protect the system against similar virus infection. In each step, our proposed model will delete the mutated self antigens from Self set in time through the dynamic description of self. The tolerance of immature detectors to mutated self antigens is thus prevented. Therefore, the false-negative error rate is reduced. Furthermore, the false-positive error rate is also reduced by adding new self antigens into Self. As the self set is dynamically defined, the immune tolerance in our model is also called dynamic tolerance.

4 Simulations and Experiments

We collected 326 distinct Windows important executable system files, which were used` as Self set. Since there is no benchmark data set available for the detection of virus. So the Non-self set (i.e. viruses) is collected from the website VX Heavens [14]. We collected 166 files from the website; include Net-Worm.Win32.Bobic, Net-Worm.Win32.Lebreat, and Net-Worm.Win32.Mytob etc.

We extracted features from files using method described in Section 3.1, to form antigen vectors. We choose 4-gram, 6-gram, 8-gram and 10-gram to form a 4-dimension vector. Initial self set was composed of 200 feature vectors of benign code. Antigen feature vectors were formed by 326 benign files and 166 malicious codes. In our experiments, we defined tolerance period of immature immune cell and lifecycle of mature immune cell, respectively, as 5 and 15. Figure 3 is the performance for detecting virus of proposed model at different self radius. The size of initial memory immune cells was set as 50. TP (True Positive) rate and FP (False Positive)

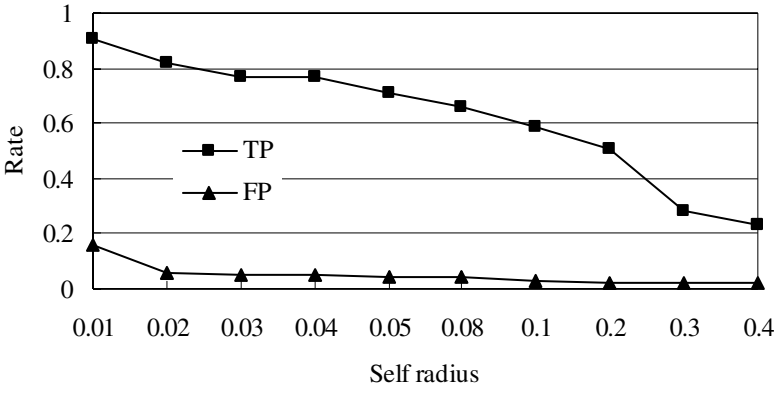


Fig. 3. TP and FP at different self radius

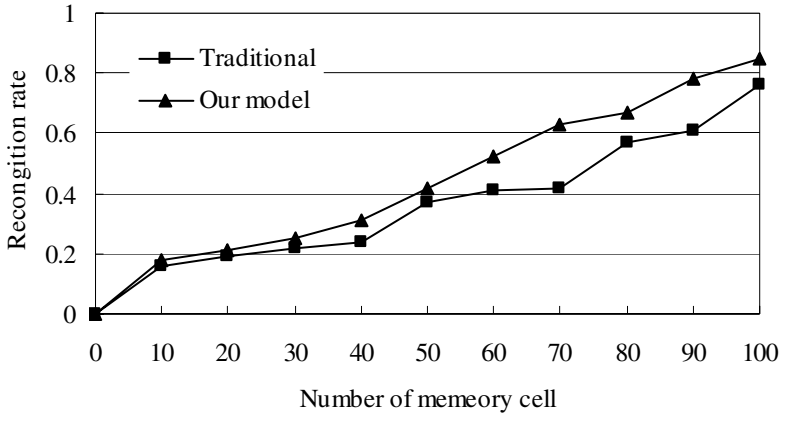


Fig. 4. Comparison of experimental results

rate denote detection rate and false detection rate, respectively. With the enlarge of self radius, benign code is less probability to be classified as virus, i.e., the FP decreased, but more virus may be regard as benign codes, that is to say, the TP decreased.

Figure.4 shows the detection rate of our model and traditional artificial immune model. The size of initial self set is set as 200; self radius is set as 0.005. In this paper, a 4 dimension vector is constructed to describe the immune cell. For the radius of detectors (mature detectors, memory detector) is variable-sized, so less mature and memory immune cells are needed to recognize lots of harmful antigens. On the other hand, time to generate detectors and to examine new samples is saved by using smaller number of detectors. It also requires less space to store them.

5 Conclusions

In traditional AIS, a large number of self cells and immune cells are required, and a great deal of time and resources is needed to generate detector cells. These shortcomings are confining the application of artificial immune system into practice. In this paper, we present a virus detection model based on the artificial immune principles with RVNS. Variable-sized detectors provide a more concise representation of the anomaly detectors derived from the benign code, time to generate detectors and to examine new samples is saved by using smaller number of detectors. A satisfied result is obtained through the principles of immune learning, recognition and memory, and etc. The experimental results show that the proposed model may have a better detection performance than traditional model.

References

1. Xu, J., Sung, A., Mukkamala, S., Liu, Q.: Obfuscated Malicious Executable Scanner. Journal of Research and Practice in Information Technology 39(3), 181–197 (2007)
2. Perelson, A.S., Weisbuch, G.: Immunology for physicists. Reviews of Modern Physics 69(4), 1219–1268 (1997)

3. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-Nonself Discrimination in a Computer. In: 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Proceedings, pp. 202–212. IEEE Computer Soc. Press, Los Alamitos (1994)
4. Kephart, J.O., Arnold, W.C.: Automatic extraction of computer virus signatures. In: 4th Virus Bulletin International Conference, pp. 178–184 (1994)
5. Kephart, J.O., Sorkin, G.B., Swimmer, M., White, S.R.: Blueprint for a Computer Immune System. In: Proceedings of the Seventh International Virus Bulletin Conference, pp. 159–173 (1997)
6. Okamoto, T., Ishida, Y.: Distributed approach against computer viruses inspired by the immune system. In: IEICE Transactions on Communications, pp. 908–915 (2000)
7. Kephart, J.O.: A Biologically Inspired Immune System for Computers. In: Artificial Life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems, pp. 130–139 (1994)
8. Gonzalez, F., Dasgupta, D., Gomez, J.: The effect of binary matching rules in negative selection. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 195–206. Springer, Heidelberg (2003)
9. Damashek, M.: Gauging Similarity with n-Grams: Language-Independent Categorization of Text. Science 267(5199), 843–848 (1995)
10. Li, T., Liu, X., Li, H.: An Immune-Based Model for Computer Virus Detection. In: Cryptology And Network Security: 4th International Conference, CANS 2005, Proceedings, Xiamen, China, December 14-16 (2005)
11. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable-sized detectors. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 287–298. Springer, Heidelberg (2004)
12. Li, T.: An immunity based network security risk estimation. Sci. China Ser. F-Inf. Sci. 48(5), 557–578 (2005)
13. Li, T.: An immune based dynamic intrusion detection model. Chinese Science Bulletin 50(22), 2650–2657 (2005)
14. VX heavens Home Page, `http://vx.netlux.org/`

A Novel Exponential Type Swarming of Foraging and Obstacle-Avoidance Behaviour Modelling and Simulating Research on Collective Motion in Multi-obstacle Environment

Zhi Bin Xue[1,3] and Jian Chao Zeng[1,2]

[1] College of Electric & Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China
zbxue_jack@163.com
[2] Division of System Simulation and Computer Application, Taiyuan University of Science and Technology, Taiyuan 030024, China
[3] Department of chemical machinery, Chemical engineering college, Qinghai University, Xining 810016, China

Abstract. A novel Lagrangian "individual-based" isotropic continuous time exponential type swarming model with a family of attraction/repulsion function is proposed in this article. The stability of aggregating behavior of the swarm systems is verified by computer numerical imitation. Numerical simulation results further indicate that the individual members living in group during the course of coordinative motion can realize the collision-eluding obstacles and ensure mutual aggregating behaviour, the motion of each individual member is a combination of the inter-individual interactions and the interaction of the member with external environment. Better self-adaptability of surrounding environment and practicability are embodied out in the proposed model.

Keywords: exponential type swarm model, collective motion, stability, numerical imitation, collision-eluding obstacles, aggregating behavior.

1 Introduction

In the natural world, population appears in patterns of aggregation such as schools of fish, flocks of birds, and herds of ungulates. The details of such aggregations are important because they influence numerous fundamental processes like predator avoidance, foraging success and so on. Recently, to study the phenomena of aggregation is an interesting topic in biology, physics and control engineering. Meanwhile, modelling and exploring the collective dynamics of swarms aggregating behavior has become an important issue in biomimicry of foraging and swarming for using in engineering applications, such as the coordinated control of multi-agent systems. However, most available results in the literature are on the isotropic non-exponential swarming model, convincing results on the exponential type swarms model are relatively few. Based on the analysis of various biological swarms of dynamic aggregation mechanism, an

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 454–460, 2008.
© Springer-Verlag Berlin Heidelberg 2008

isotropic exponential type swarming dynamic model is proposed in this paper. Dynamic change of the environment, local observation and nonlinear characteristics are ubiquitous phenomena in nature, but the study is very difficult and it has profound engineering significance.

In the individual-based (or Lagrangian) frameworks models, the basis rule is that the swarming behavior is a result of an interplay between a short-ranged repulsion and a long-ranged attraction between the individuals and interly with environment [1]. Models which use forces between individuals that are analogous to physical forces, have a long history. Parr proposed the idea of mutual interactions between individuals that are composed of attractions and repulsion, designed to maintain the group as a stable mass [2]. Breder developed this idea further by discussing the possibility of modelling attraction forces between individual fish upon classical gravitation and electromagnetism [3]. In a later paper, Breder considered inverse power laws to model the repulsions and attractions between individuals, with repulsion stronger at short inter-individual distances. Breder compared this model to actual fish schools, to obtain realistic values for model parameters. This article is one of the very first instances of the use of attraction and repulsion biological forces, Breder proposed a simple model composed of a constant attraction and a repulsion inversely proportional to the square of the inter-individual distance [4]. In this paper, a family of attraction/repulsion-distance functions is used to construct exponential type swarming model. Computer simulation was used to examine the effects of each function on group cohesion, as reflected by foraging and obstacle-avoidance behavior. There has been a strong effort both on computational and mathematical front to model animal group behavior. The implications for aggregate animal behavior extend far beyond an understanding of animal behavior. The same laws governing animal behavior can be generalized to guide artificial machines or robots to carry out complex routines by relying only on local interactions. So this article is able to provide some results on this topic.

The paper is organized as follows. In the next section we present a new individual-based continuous-time isotropic exponential swarm model. A family of attraction-repulsion function that can avoid collisions and numerical imitation analyze the collective motion under the interplays of the swarm individuals which the affect of the external environment attractant/repellent profile is considered is given in Section 3. In Section 4, we extend the results in Section 3 by considering swarms foraging and obstacle-avoidance behavior of collective motion in multi-obstacle environment, in order to reify our theories. We conclude our results in Section 5.

2 The Isotropic Exponential Type Swarm Model Description

Mogilner, Edelstein-Keshet, Bent & Spiros propose a Lagrangian model based on attractive and repulsive potentials, to investigate the spacing of individuals in a social aggregate. This is a different approach to an earlier paper, which uses an Eulerian

approach. Their attractive and repulsive potential functions are formed using both exponential and power laws. The authors show (analytically and numerically) that repulsion terms must dominate attraction potentials at a short range for a well spaced biologically plausible group to form. Liapunov functions are used to evaluate equilibrium position resulting from their models [5].

Swarm model with isotropic exponential forces property can adapt well to environment and follow the principles of short-ranged repulsion and long-ranged attraction which are needed for realistic spacing behavior and cohesion in a group. Refering to the known results in literatures [5], [6] and [7], we consider a swarm of *M* individuals (members) in a *n*–dimensional Euclidean space, assume synchronous motion and no time delays, and model the individuals as points and ignore their dimensions. The equation of collective motion of individual *i* is given by as followes

$$\dot{x}^i = h_{x^i} + \sum_{j=1, j\neq i}^{M} g\left(x^i - x^j\right), i = 1, \cdots, M \ . \tag{1}$$

Where $x^i \in R^n$ represents the position of individuals i; h_{x^i} stands for the effect in heading direction resting with social potential fields around individual i; $g(\cdot)$ represents the function of attraction and repulsion between the members .

The attraction/repulsion function that we consider is

$$g(y) = sign(y) \cdot \left[A \cdot \exp\left(-\frac{\|y\|}{a} \right) - R \cdot \exp\left(-\frac{\|y\|}{r} \right) \right]. \tag{2}$$

Where the parameters R, A>0 are magnitudes of attraction and repulsion and r, a>0 are the spatial ranges of the attraction and repulsion. While R>A, r>a, and $R \cdot r^2 \succ A \cdot a^2$, this is the short-ranged repulsion and long-ranged attraction case, and is the most interesting and biologically relevant [5]. 2-norm $\|y\| = \left\|x^i - x^j\right\| = y^T y$ represents the distance between the individual i and individual j. Due to the fact that $g(\cdot)$ are odd functions, the convergence properties of the swarm to minimum (or critical) points of the profile depends on the properties of the profile [1].

Where $h_{x^i} = -\nabla \sigma\left(x^i\right)$, the parameter h_{x^i} stands the motion of the individuals toward destination regions and away from obstacles regions .

3 Numerical Imitation Results of Collective Motion

Based on the same simulation method, process and parameters given in reference [6], includes Plan, Quadratic, Gaussian, Multimodal Gaussian attractant/repellent social potential field profiles functions, we performed simulations, numerical imitation results as follows

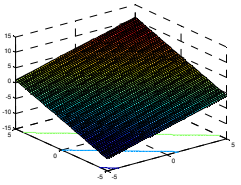


Fig. 1. Plan social potential field

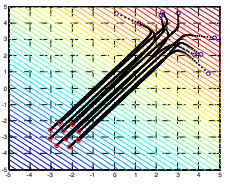


Fig. 2. Collective behavior of swarm in plane social potential field in reference [6]

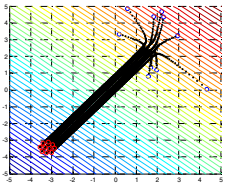


Fig. 3. Collective behavior of swarm in plane social potential field of the exponential type swarm model

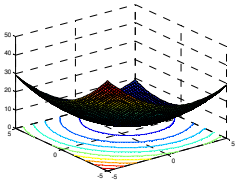


Fig. 4. Quadratic social potential field

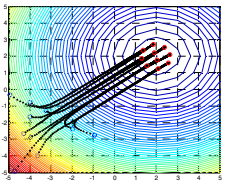


Fig. 5. Collective behavior of swarm in quadratic social potential field in reference [6]

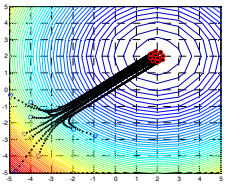


Fig. 6. Collective behavior of swarm in quadratic social potential field of the exponential type swarm model

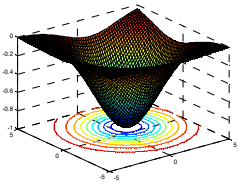


Fig. 7. Gaussian social potential field

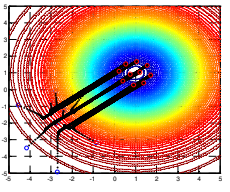


Fig. 8. Collective behavior of swarm in Gaussian social potential field in reference [6]

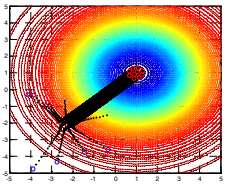


Fig. 9. Collective behavior of swarm in Gaussian social potential field of the exponential type swarm model

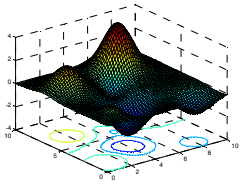


Fig. 10. Multimodal Gaussian social potential field

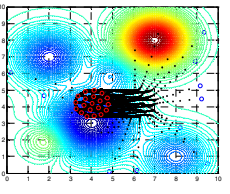


Fig. 11. Collective behavior of swarm in Multimodal Gaussian social potential field in reference [6]

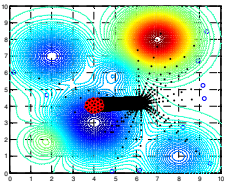


Fig. 12. Collective behavior of swarm in Multimodal Gaussian social potential field of the exponential type swarm model

The theoretical analysis and simulation results in this paper confirm that the convergence properties of the exponential type swarm model are better than the model in reference [6].

4 Further Extensions

In this section, we consider swarms obstacle-avoidance behavior of collective motion in multi-obstacle environment.

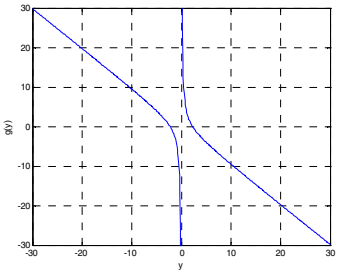


Fig. 13. The isotropic exponential type swarm model attraction/repulsion function $g(\cdot)$

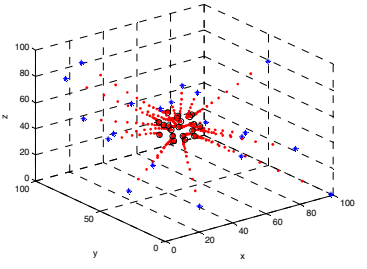


Fig. 14. Convergent trajectories of individuals in swarm model in reference [6]

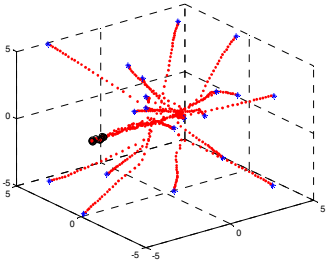


Fig. 15. Convergent trajectories of individuals in isotropic exponential type swarm model

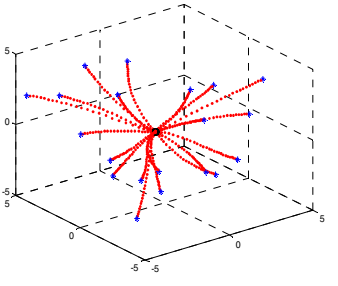


Fig. 16. Convergent trajectories of individuals in isotropic exponential type swarm model and we used the $\tanh(\cdot)$ function instead of the $sign(\cdot)$ function to smooth the motion trajectories

By comparison, the convergence properties of Fig. 16 is better than Fig. 14. Blue "*" represent original position, black "。" represent final position, read "." Represent convergent trajectories of individuals.

Based on the same simulation method, and process in reference [7], by using the isotropic exponential type swarm model, we obtain the following simulation results as shown in Fig. 17 is for 100 individuals to traverse through an environment with five obstacles. Where the red ball's center represent the global object position is at (78 cm, 78

cm) and the simulation region is 80 cm × 80 cm in the space. The five black balls' center is repectively: (35 cm, 45 cm), (48 cm, 35 cm), (52 cm, 50 cm), (50 cm, 45 cm) and (65 cm, 65 cm). The five black balls represents the obstacles in the environment. The yellow dots represent the convergent trajectories of individuals in swarm. The result shown in Fig. 17 reify our theories, in multi-obstacle environment, the individuals in the swarm during the course of coordinative motion can realizes the collision-eluding obstacles, mutual aggregating behavior and arrive at object position finally.

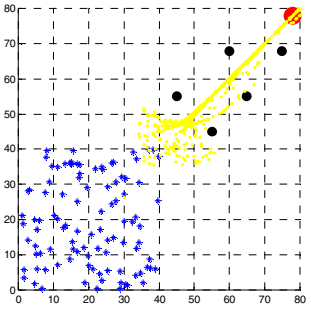


Fig. 17. The track of the swarm in multi-obstacle environment

5 Conclusions

In this paper, based on the inspiration from biology, we consider an isotropic exponential type swarm model of aggregating for multi-agent system. The model is a kinematic model, meanwhile, it is fit for individuals which move basing on the Newton's law in an environment can capture the basic convergence properties of biological populations in nature. Therefore, the final behavior of the swarms described by the model may be in harmony with real biological swarms well. The simulation results in this paper confirm that the stability of aggregating behavior of the swarm systems. As space is limited, this article can't included every party involved, only a brief account of the basic idea and method of the stability theoretical analysis is presented here. Furthermore, the aggregating results obtained in this model which has a definite reference value in the multi-agent coordination and control literature. Numerical simulation experiments show that the model can guarantee collision avoidance in the swarm in multi-obstacle environment.

References

1. Gazi, V., Passion, K.M.: Stability Analysis of Social Foraging Swarms. IEEE Trans. Systems, Man, and Cybernetics 34(1), 539–557 (2004)
2. Parr, A.E.: A contribution to the theoretical analysis of the schooling behaviour of fishes. Occasional Papers of the Bingham Oceanographic Collection 1, 1–32 (1927)

3. Breder, C.M.: Structure of a fish school. Bull. Amer. Mus. Nat. Hist. 98, 1–27 (1951)
4. Breder, C.M.: Equations descriptive of fish schools and other animal aggregations. Ecology 35, 361–370 (1954)
5. Mogilner, A., Edelstein-Keshet, L., Bent, L., Spiros, A.: Mutual interactions, potentials, and individual distance in a social aggregation. J. Math. Biol. 47, 353–389 (2003)
6. Chen, L., Xu, L.: Collective behavior of an anisotropic swarm model based on unbounded repulsion in social potential fields. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS (LNBI), vol. 4115, pp. 164–173. Springer, Heidelberg (2006)
7. Chen, S.M., Fang, H.J.: Modeling and stability analysis of social foraging swarms in multi-obstacle environment. Journal of Control Theory and Applications 4, 343–348 (2006)

Algorithm of On-Line Handwriting Signature Verification Based on Discrete Fréchet Distance

Jianbin Zheng[1], Xiaolei Gao[1], Enqi Zhan[1], and Zhangcan Huang[2]

[1] School of Information Engineering, Wuhan University of Technology, China
[2] School of Science, Wuhan University of Technology, China
{zhengjb@whut.edu.cn,gxltcl1985}@yahoo.com.cn,
eqzhan@whut.edu.cn,huangzc@whut.edu.cn

Abstract. A new algorithm for on-line handwriting signature verification is proposed. The algorithm extracts the position coordinates of extreme points of reference signature and test signature in the signature curves, and then uses discrete Fréchet distance as the measure of the curve distance, respectively matching peak points and peak points, and valley points and valley points. Finally, a decision is made to see if the test signature is genuine. A new definition of curve similarity is introduced in the algorithm and a new mathematical model of judging signature curve similarity has been built up on this definition. The algorithm implies shifting and stretching transformation of the signature curves. The experimental results show the efficiency and feasibility of this algorithm.

Keywords: Signature verification, discrete Fréchet distance, matching, decision-making.

1 Introduction

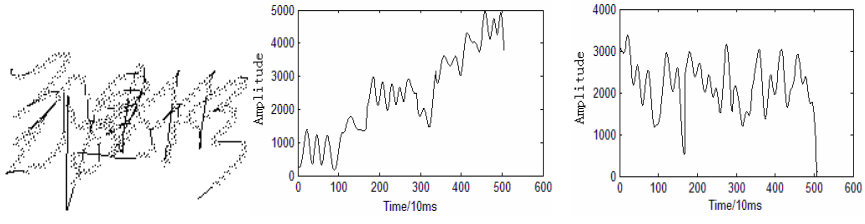
With the continuous development of science and technology, the popularity of computers and networks make us more convenient in communication and contact, but the information security issues are increasingly prominent. Thus it is an essential issue of how to identify accurately a person's identity and protect information security in this information era. At present, iris, fingerprint, voice and handwriting signature are widely used in the field of identity identification. Comparing with other identity verification technology, the handwriting signature verification technology costs less, and does not require specialized and complex sampling equipment. The signature is more unforgettable than the traditional password verification technology. Therefore, the signature verification technology will have wide prospects in the field of information security.

Since the 1980s, many scholars have committed themselves to the research on the algorithms of signature verification. The algorithm of signature verification mainly contains two types: the method of parameter features and the method of function features. There are some more typical algorithm as follows: Darwish and Auda used neural network as classifiers[1], and investigated total 210 signature features which were found by the predecessors. Mohankrishnan and Paulik proposed a signature verification method which was based on the Autoregressive model[2]. And Yang and

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 461–469, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Widjaja proposed hidden Markov model method[3]. Connell proposed a matching method based on feature vector[4], and maintained that feature selection should satisfy the requirement of simple and easy calculation, having nothing to do with shifting, rotation and stretching. Nakanishi introduced the wavelet analysis method in the on-line signatures verification[5], and decomposed signature signal into 8 levels. During the 1990s, some researchers used the signature energy features of sharp trajectory change points in the signature signal as features[6]. Recently Zheng Jianbin presented a signature verification algorithm based on evolutionary computation[7]. The test signature curve matches with the reference signature curve by self-adaptive segmentation. These methods greatly push forward the technology of signature verification. From another perspective, this paper tries to introduce discrete Fréchet distance into signature verification.

On-line signature verification system real-time collects handwriting signature signal through PC tablet. The PC tablet can not only collect signatures position coordinates, but also record writing speed, pressure and other dynamic information. Usually, we can gather X coordinate signal, Y coordinate signal and pressure value. Fig.1 is ZJB's some signature and its position coordinate curves. Whether test signature matches with reference signature depends on the comparison of test signature coordinate curves and reference signature coordinate curves.



(a) signature (b) X coordinate (c) Y coordinate

Fig. 1. ZJB's signature with its collected position coordinate signals. Fig.1 (a) is ZJB's signature. Fig.1 (b) is the collected X coordinate. Fig.1 (c) is a collected Y coordinate.

However, owing to the random of on-line signature, the function which expresses a signature is very complicated. Therefore, all the present methods of feature extraction and comparison are far from satisfactory. It is difficult to propose a better rule to judge the similarity of reference and test signature curves. Accordingly it causes great difficulties to compare reference signature and test signature. This paper gives a new definition of the curve similarity and proposes a new algorithm of on-line handwriting signature verification. The algorithm extracts peak points and valley points of the signature curves, and compares the similarity of reference and test signature curves by using discrete Fréchet distance[8] as an evaluating criteria. A mathematics model has been built up on this algorithm. The model implies shifting and stretching transformation of the signature curves.

2 Mathematical Model

The mathematical definition of discrete Fréchet distance is shown as follows:

Def 1[9]: The Fréchet distance δ_F between two parametric curves: $f:[0,1]\to R^2$ and $g:[0,1]\to R^2$ is defined as follows:

$$\delta_F(f,g)=\inf_{\alpha,\beta}\max_{s\in[0,1]} dist(f(\alpha(s)),g(\beta(s)))\,. \tag{1}$$

Where α and β range over all continuous non-decreasing real functions with $\alpha(0)=\beta(0)=0$ and $\alpha(1)=\beta(1)=1$.

Def 2[10]: 1.Given a polygonal chain $P=<p_1,p_2,...,p_n>$ of n vertices, a k-walk along P partitions the vertices of P into k disjoint non-empty subsets $\{P_i\}_{i=1...k}$ such that $P_i=<p_{n_{i-1}+1},...,p_{n_i}>$ and $0=n_0<n_1<...<n_k=n$.

2. Given two polygonal chains $A=<a_1,...,a_m>$ and $B=<b_1,...,b_n>$,a paired walk along A and B is a k-walk $\{A_i\}_{i=1...k}$ along A and a k-walk $\{B_i\}_{i=1...k}$ along B for some k, such that, for $1\le i\le k$ either $|A_i|=1$ or $|B_i|=1$ (that is, either A_i or B_i contains exactly one vertex).

3. The cost of a paired walk $W=\{(A_i,B_i)\}$ along two chains A and B is

$$d_F^W(A,B)=\max_i\ \max_{(a,b)\in A_i\times B_i} dist(a,b)\,. \tag{2}$$

The discrete Fréchet distance between two polygonal chains A and B is

$$d_F(A,B)=\min_W d_F^W(A,B)\,. \tag{3}$$

The paired walk $W=\{(A_i,B_i)\}$ that achieves the discrete Fréchet distance between two polygonal chains A and B is called the Fréchet alignment of A and B.

For the definition of discrete Fréchet distance merely examines the distance of peak points or valley points, it is far insufficient to judge the curve similarity. Thus we introduce a new curve similarity definition in Def 3.

Def 3: Given two polygonal chains $A=<a_1,...,a_m>$ and $B=<b_1,...,b_n>$, which are composed of discrete points, the discrete Fréchet distance of peak points $d_F(A,B)$ is called $d_F^1(A,B)$, and the discrete Fréchet distance of valley points $d_F(A,B)$ is called $d_F^2(A,B)$. For a given value ε, if $\left|d_F^1(A,B)-d_F^2(A,B)\right|\le\varepsilon$, we consider that A is silimar to B, otherwise they are not similar.

3 Design and Implementation of the Algorithm

3.1 Preprocessing of Signature Curve and Extraction of Extreme Points

Preprocessing such as elimination of zero points, flying points, and the leaking points is carried out in the primary signature curves, and then weighted smooth filtering is processed so that the stable extreme points can be extracted.

The method of extracting stable extreme points is as follows: as for $a(n)$, which is a point in signature curve, if $a(n) > a(n-1)$, $a(n) > a(n+1)$, $a(n-1) > a(n-2)$, and $a(n+1) > a(n+2)$, then we can think that $a(n)$ is a stable peak point; and if $a(n) < a(n-1)$, $a(n) < a(n+1)$, $a(n-1) < a(n-2)$, and $a(n+1) < a(n+2)$, then we can think that $a(n)$ is a stable valley point.

All of stable peak points and valley points in a certain subject's signature curve are shown in Fig.2.

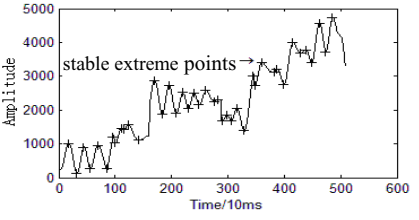


Fig. 2. Stable extreme points in the X coordinate curve of a certain signature

3.2 Discussion about Search Space

Given m peak points are extracted in reference signature curve, they are recorded as one set: $A = \{a_1, a_2, \ldots, a_m\}$, while n peak points are extracted in test signature curve, and recorded as the other set: $B = \{b_1, b_2, \ldots, b_n\}$, and moreover, $m \le n$. Dividing the set B into m parts with the Fréchet alignment definition, we suppose that there are k kinds of division methods, where $k \in R$. The j^{th} alignment of $A\ and\ B$ is $W_j = \{(A_i, B_i)\}, (1 \le i \le m, 1 \le j \le k)$. For $1 \le i \le m$, the algorithm stipulates $|A_i| = 1$ (that is, each parts of set A exactly contains one peak point). And if $A_i =< a_i >$, $B_i =< b_i, b_{i+1} >$, then we get $A_{i+1} =< a_{i+1} >$, $B_{i+1} =< b_{i+2} \ldots >$. That is to say, the present division influences the next division.

If two curves are similar, then i^{th} peak point of reference signature curve is only possibly correlated with the correspondence peak point or its neighbor peak point of test signature curve. So in this algorithm the search space will be set as follows: i^{th} peak point of reference signature curve is only correlated with the i^{th} peak point, $i-1^{th}$ peak point, $i+1^{th}$ peak point or not correlated with any peak points in test signature. To sum up, it is implied that some extreme points will be moved or deleted in the signature curve.

Take Fig.3 as an example. There are 10 peak points in reference signature curve recorded as $A = \{a_1, a_2, \ldots, a_{10}\}$, while there are 11 peak points in test signature curve recorded as $B = \{b_1, b_2, \ldots, b_{11}\}$. Fig.3 is a kind of Fréchet alignment of set $A\ and\ B$ in k kinds of Fréchet alignments arranged as follows:

$A_1 =< a_1 >, B_1 =< b_1 >$; $A_2 =< a_2 >, B_2 =< b_2 >$; $A_3 =< a_3 >, B_3 = 0$;

$A_4 =< a_4 >, B_3 =< b_3 >$; $A_5 =< a_5 >, B_5 =< b_4 >$;

$A_6 =< a_6 >, B_6 =< b_5, b_6, b_7 >$; $A_7 =< a_7 >, B_7 =< b_8 >$;

$A_8 =< a_8 >, B_8 =< b_9 >; A_9 =< a_9 >, B_9 =< b_{10} >; A_{10} =< a_{10} >, B_{10} =< b_{11} >$

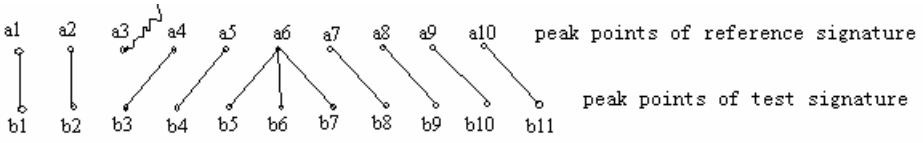


Fig. 3. A kind of Fréchet alignment in k kinds of Fréchet alignments

3.3 Algorithm of Curve Similarity Based on Discrete Fréchet Distance

The core algorithm of signature verification based on discrete Fréchet distance is shown as follows:

Step 1: m peak points, extracted in reference signature curve, are compared with n peak points in test signature curve. If $n - m \geq 5$, then we think that reference signature is not similar to test signature. Accordingly the algorithm is finished; otherwise the algorithm turns to step 2;

Step 2: Divide n peak points in set B into m parts. According to the Fréchet alignment definition and the scope of the search space, we can find out all of the division methods. Suppose that there are k kinds of divisions ($k \in R$), each division or Fréchet alignment $of\ A\ and\ B$ is $W_j = \{(A_i, B_i)\}, (1 \leq i \leq m, 1 \leq j \leq k)$. Each part of set B will calculate the distance with each peak point of set A . And the distance between peak points is defined as follows: $d = | a_i - b_j |$, $(1 \leq i \leq m, 1 \leq j \leq n)$. The distance between each part of extreme points in two signatures will be calculated in pairs, and the maximum distance of each part inside is chosen. According to the definition of *paired walk,* we figure out the maximum one in all the walks, that is $d_F^W(A,B) = \max_i \max_{(a,b) \in A_i \times B_i} dist(a,b)$.

Step 3: Find out the superior division, or the minimum distance of all the division. $d_F(A,B) = \min_W d_F^W(A,B)$ is the discrete Fréchet distance of A and B ;

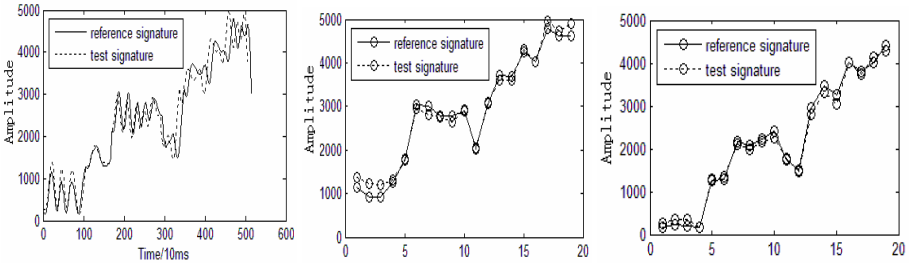
Step 4: Calculating the discrete Fréchet distance between peak points and peak points in reference signature and test signature curves, we can get $d_F^1(A,B)$. Calculating the discrete Fréchet distance between valley points and valley points in reference signature and test signature curves, we can get $d_F^2(A,B)$. For a given threshold ε , if $\left| d_F^1(A,B) - d_F^2(A,B) \right| \leq \varepsilon$, then we can conclude that reference signature is similar to test signature; otherwise they are not similar. The smaller $d_{\min} = \left| d_F^1(A,B) - d_F^2(A,B) \right|$ is, the more similar the reference signature curve and the test signature curve are.

4 Experimental Result and Analysis

For convenience, all the figures below are taken for examples by X coordinate curve. By comparison of ZJB's genuine signatures, random forged signatures and skilled forged signatures, we can set the X coordinate threshold as follows: $\varepsilon = 150$; If $d_{\min} = \left| d_F^1(A,B) - d_F^2(A,B) \right| \le \varepsilon$, we can conclude that reference signature and test signature curves are matched.

Everyone has different habit and styles of writing in the signature, and has different thresholds. The process of confirming everyone's threshold is that of lasting training and learning his own signatures. With the updating of data in the signature sample base, every individual's threshold has to be modified through training and learning.

The reference signature and test signature are ZJB's genuine signatures in signature sample base, and X coordinate curves, extracted peak points and valley points are shown in Fig.4.



(a) X coordinate curves (b) extracted peak points (c) extracted valley points

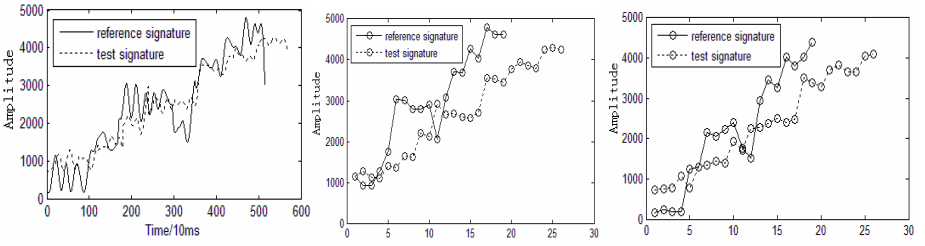
Fig. 4. X coordinate curves of two genuine signatures in Fig.4 (a); Extracted peak points in Fig.4 (b); Extracted valley points in Fig.4 (c)

There are someone's several genuine signatures chosen in the signature sample in table 1. Then the superior value $d_{\min}$ is obtained. By comparison of $d_{\min}$ with the threshold ε, judgment is made.

Tabl 1. ZJB's experimental results of genuine signature matching

Reference signatures	Test signatures	$d_{\min}$	Matched?
zjb01.h	zjb02.h	|383-237|=146	Yes
zjb01.h	zjb03.h	|245-278|=33	Yes
zjb01.h	zjb04.h	|383-506|=123	Yes
zjb01.h	zjb05.h	|302-364|=62	Yes
zjb01.h	zjb06.h	|362-247|=115	Yes
zjb01.h	zjb07.h	|564-596|=32	Yes
zjb01.h	zjb08.h	|279-250|=29	Yes

Reference signature is ZJB's genuine signature, and test signature is random forged signature. X coordinate curve, extracted peak points and valley points are shown in Fig.5.



(a) X coordinate curves (b) extracted peak points (c) extracted valley points

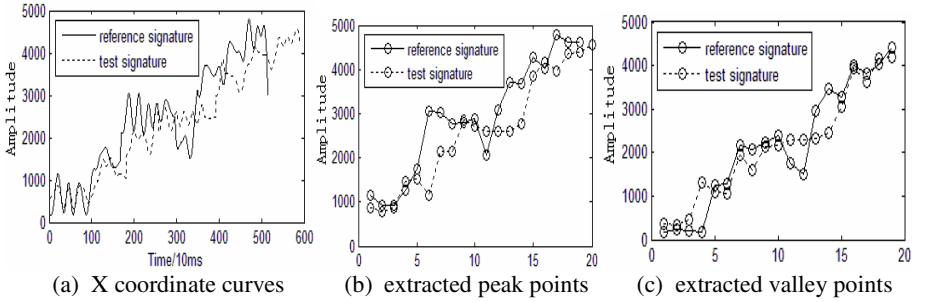
Fig. 5. X coordinate curves of genuine signature and random forged signature in Fig.5 (a); Extracted peak points in Fig.5 (b); Extracted valley points in Fig.5 (c)

There are ZJB's genuine signatures and random forged signatures chosen in the signature sample base in table 2. Then the superior value $d_{\min}$ is obtained. By comparison of $d_{\min}$ with the threshold ε , judgment is made.

Table 2. ZJB's experimental results of genuine and random forged signature matching

Reference signatures	Test signatures	$d_{\min}$	Matched?
zjb01.h	zjb_gxl01.h	|1078-735|=343	No
zjb01.h	zjb_gxl02.h	|464-717|=253	No
zjb01.h	zjb_gxl03.h	|680-792|=170	No
zjb01.h	zjb_gxl04.h	|889-648|=241	No
zjb01.h	zjb_gxl05.h	|974-725|=249	No
zjb01.h	zjb_gxl06.h	|664-840|=176	No
zjb01.h	zjb_gxl07.h	|1527-850|=677	No

Reference signature is ZJB's genuine signature, and test signature is skilled forged signature. X coordinate signal, extracted peak point curves and valley point curves are shown in Fig.6.



(a) X coordinate curves (b) extracted peak points (c) extracted valley points

Fig. 6. X coordinate curves of genuine signature and skilled forged signature in Fig.6 (a); Extracted peak point curves in Fig.6 (b); Extracted valley point curves in Fig.6 (c)

There are ZJB's genuine signatures and skilled forged signatures chosen in the signature sample base in table 3. Then the superior value $d_{\min}$ is obtained. By comparison of $d_{\min}$ with the threshold ε , judgment is made.

Table 3. ZJB's experimental results of genuine and skilled forged signature matching

Reference signatures	Test signatures	$d_{\min}$	Matched?
zjb02.h	zjb_wlj01.h	|756-934|=178	No
zjb02.h	zjb_wlj02.h	|928-738|=190	No
zjb02.h	zjb_wlj03.h	|905-689|=216	No
zjb02.h	zjb_wlj04.h	|1283-1085|=198	No
zjb02.h	zjb_wlj05.h	|949-789|=160	No
zjb02.h	zjb_wlj06.h	|604-632|=28	Yes
zjb02.h	zjb_wlj07.h	|764-461|=303	No

By comparison of the same person's signatures during different periods in signature sample base, the FRR(Fault Rejection Rate) is about 1.8%. By comparison of genuine signatures and random forged signatures in signature sample base, the FAR(Fault Acceptance Rate) is about 5%. By comparison of genuine signatures and skilled forged signatures, FAR is about 10.1%.

According to the above experimental results, the used method based on discrete Fréchet distance can well judge the random forged signatures and skilled forged signatures, and it can also accurately verify genuine signatures. The algorithm has a higher verification rate for the genuine signatures, and a lower FAR for the random forged signatures.

5 Conclusion

In this paper, we apply discrete Fréchet distance to on-line handwriting signature verification, and present a new algorithm of judging signature similarity. By analysis of the experimental result, the algorithm quickens the speed of signature verification, and improves the operational results of the signature verification. Experimental results reveal the feasibility of this algorithm. We just consider coordinate position features of extreme point. Therefore, this algorithm is very efficient.

But there are still some problems to be solved. At present, the algorithm has somewhat higher FAR for some skilled forged signatures in the matching process. Therefore, the algorithm performance may be improved by addition of the speed features and some other dynamic features.

References

1. Darwish, A.M., Auda, G.A.: A New Composite Feature Vector for Arabic Handwritten Signature Verification. In: Procceedings of IEEE International Conference on Acoustics, vol. 2, pp. 613–616 (1994)
2. Mohankrishnan, N., Paulik, M.J., Khalil, M.: On-line Signature Verification Using a Non-stationary Autoregresssive Model Representation. In: IEEE International Symposium on Circuits and Systems, vol. 7, pp. 2303–2306 (1993)

3. Yang, L., Widjaja, B.R., Prasad, R.: Application of Hidden Markov Model for Signature Verification. Pattern Recognition 2, 180–188 (1996)
4. Connell, S.D.: On-line Handwriting Recognition Using Multiple Pattern Class Models. Ph.D. thesis, MSU-CSE-00-27, Michigan State University (2000)
5. Nakanishi, I., Nishiguchi, N., Itoh, Y.: On-line Signature Verification Method Utilizing Feature Extraction Based on DWT. IEEE Transactions on Information Theory 2, 691–712 (2003)
6. Zheng, J., Zhu, G.: On-line Handwriting Signature Recognition Based on Wavelet Energy Feature Matching. In: IEEE Proceedings of the World Congress on Intelligent Control and Automation (WCICA), pp. 9885–9888 (2006)
7. Zheng, J., Zhu, G.: New Algorithm for On-line Handwriting Signature Verification Based on Evolutionary Computation. Wuhan University Journal of Natural Sciences 3, 596–600 (2006)
8. Jiang, M., Xu, Y., Zhu, B.: Protein Structure-Structure Alignment with Discrete Fréchet Distance. In: Web of science, ISI proceedings, pp. 1–12 (2007)
9. Alt, H., Godau, M.: Measuring the Resemblance of Polygonal Curves. In: Proceedings of the 8th Annual Symposium on Computational Geometry (SoCG 1992), pp. 102–109 (1992)
10. Eiter, T., Mannila, H.: Computing Discrete Fréchet Distance. Technical Report CD-TR94/64, Technical University of Vienna (1994)

Dynamic Model and Numerical Simulation of Particle Motion in Rotation Flow Field of Centrifuge

Tong Zhu, Tsutomu Nozaki, Yuanhua Xie, Jin Han, Jing Jiang, and Bing Li

School of Mechanical Engineering and Automation
Northeastern University, Shenyang, Liaoning, 110004, China
tongzhu@mail.neu.edu.cn

Abstract. In this study, in order to determine the size, configuration, rotating speed and other parameters of centrifuge separator, particle motion was simulated by the method of numerical integration of Runge-Kutta. For setting up of dynamic model, besides of forces of gravity, buoyancy, drag force and centrifugal force, the forces of imagination mass force, Basset force, Saffman lift force and pressure gradient force were considered. As the results, it was found that there were some relations among the length of drum, the particle diameter and the revolution of drum. There exist much differences of particle motion between different particles in various sizes. The visualization experiments were carried out to verify the calculated results. The experiment and calculation results were suited well.

1 Introduction

It is difficult to separate solid particles smaller than 0.1mm from liquid by the conventional cyclone separator, because of the density difference between liquid and solid being order of 100 whereas 103 in the case of solid-gas flow. A new type of the solid-liquid separator called a Tapered Drum Rotating Type Separator (TDRS) was proposed by the authors (1,2). It can efficiently separate smaller particles from the solid-liquid two-phase mixture. By rotating the drum, there exists a relative rest inside the drum filled with water. Therefore, even small solid particles can be separated from the liquid by means of the dominant centrifugal force acting upon the particle.

In the previous papers, as one of the fundamental studies on the separator, both numerical calculation and the experiment of the particle motion in the straight cylindrical drum were carried out to verify the calculated results. In this paper, in order to obtain more precise information about the practical TDRS, the numerical simulation of motion of a solid particle thrown into the relative rest field was carried out using very small particles and high revolution of drum, and carried out for different density of particle. Furthermore, the top angle of drum and the vertical downward velocity were taken into consideration in this numerical simulation.

In order to transfer the particles along the wall downward continuously, the top angle of drum is desirable to make as large as possible. However, the top angle of drum and the height of drum are desirable as small as possible, because of the bottom radius of drum being to be as smaller as possible at the viewpoint of sealing, cost and safety of the system. As the first step of this calculation, the top angle of drum was kept

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 470–478, 2008.
© Springer-Verlag Berlin Heidelberg 2008

constant at 20°which was determined from the previous experiment using the practical TDRS.

The calculations were also carried out by adding the vertical downward velocity to the relative rest field, which strongly relates to the treatment rate of solid-liquid flow, for example, a waste water treatment, as the performance of the separator. The particle diameter and the rotation of drum were also varied in the calculation. The aim of this paper is to develop previous calculation carried out by the authors (3) in order to obtain more precise information concerning the TDRS.

2 Calculation Method

Fig.1 shows the schematic model of the forces acting upon a particle in a rotating tapered drum which has the top inclined angle and filled with fluid. The equations of particle motion are almost similar as deduced by the previous paper (3), except the drum shape being different and the vertical velocity Vo being taken into consideration in this calculation, so the calculation method are briefly shown here.

The equation of motion of a particle is given by

$$m(\mathrm{d}\overline{V_p}/\mathrm{dt}) = \sum \overline{F_i} \tag{1}$$

Where m denotes the mass of particle, Vp the particle velocity and F_i the forces acting upon a particle including the gravity force F_1, the buoyancy force F_2, the drag force F_3, the apparent mass force F_4, the Basset force F_5, the Saffman lift force F_6(4) and forces based on the pressure gradient F_7 and the centrifugal force F_8.

The most important forces as gravity force F_1 and buoyancy force F_2 in the direction of gravity are given by

$$\overline{F_1} = m_p g = V\rho_p \overline{g} \tag{2}$$

$$\overline{F_2} = m_f g = V\rho_f \overline{g} \tag{3}$$

The motion of particle in the direction of gravity is due to these two forces.

A steady-state drag force F_3 is given by

$$\overline{F_3} = (1/2)\rho_f C_D A\left|\overline{U_f} - \overline{U_p}\right|\left(\overline{U_f} - \overline{U_p}\right) \tag{4}$$

The unsteady force named the virtual mass effect F_4 and Basset force F_5 are given by

$$\overline{F_4} = C_v(\rho_f V/2)\left(\mathrm{d}\overline{U_f}/\mathrm{dt} - \mathrm{d}\overline{U_p}/\mathrm{dt}\right) \tag{5}$$

And

$$\overline{F_5} = C_B(3/2)d_p^2\sqrt{\pi\rho_f\mu}\cdot \left[\int_0^t (\mathrm{d}\overline{U_f}/\mathrm{dt} - \mathrm{d}\overline{U_p}/\mathrm{dt})/\sqrt{t - t'}\,\mathrm{dt} + (\overline{U_f} - \overline{U_p})_{t=0}/\sqrt{t}\right] \tag{6}$$

which are caused by the accelerating motion of particles in fluid.

The lift force named the Saffman lift force F_6 is given by

$$\overline{F_6} = (1/2)C_L \rho_f A|\overline{U_f} - \overline{U_p}|(\overline{U_f} - \overline{U_p}) \tag{7}$$

It is due to the pressure distribution developed on a particle due to rotation induced by a velocity gradient. In this paper, the direction of Saffman force is towards the center of the rotating drum which having inclined angle.

A force named force based on the pressure gradient F_7 is given by

$$\overline{F_7} = \rho_f V\left((\mathrm{D}\overline{U_f}/\mathrm{Dt}) - (\mu/\rho_f)\nabla^2\overline{U_f}\right) \tag{8}$$

It is due to the pressure gradient in the fluid.

In this paper, the cylindrical coordinate is used to deal with these equations. In addition the centrifugal force F_8 is given by

$$F_8 = (m_p - m_f)(v_\theta^2/r) \tag{9}$$

Accounting for treatment of solid-liquid two-phase flow rate Q continuously, the vertical velocity $V(z)$ is taken into consideration and is given by

$$\overline{v}_{(z)} = Q/S_{(z)} \tag{10}$$

Where $S(z)$ denotes the flow sectional area. The initial vertical velocity at the entrance of drum is denoted by V_0 .

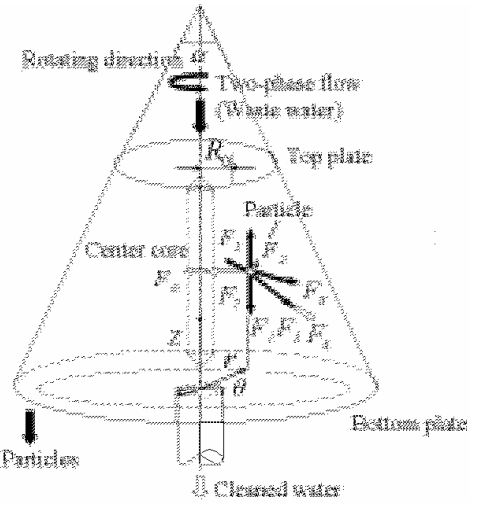


Fig. 1. Forces acting upon particle in rotating tapered drum

After reaching at the wall, the particles should go down along the wall to be able to separate the particles from water continuously. From the viewpoint of sealing, cost and safety of the system, the bottom diameter of drum is desirable as smaller as possible, so the top angle of drum should be decided as the smallest value to go down along the wall. In this calculation, the top angle of drum is given as 20°which is the smallest angle going down along the wall.

In order to judge whether the particle already reached at the wall of drum, the following inequality was taken into consideration,

$$r > Z_r \tan(\alpha/2) + R_t \tag{11}$$

Where r denotes the distance from the center axis of drum. Z_r denotes the vertical length from the starting point of particle.

Actually, as the particles flow in to avoid the center of drum which is the singular point (5), the position where a particle is thrown in is given at 10mm from the center axis of the drum.

3 Results and Discussion

In order to clarify the particle motion in the tapered drum filled with water, the following conditions of calculation for the numerical simulation were used: the density of fluid ρ_f=999kg/m^3(water), the top angel of drum α= 20°, the top diameter of drum D_t=78mm (reference length), the spherical particle diameter d_p is varied from 0.005mm to 5mm. The density of particle ρ_p is varied from 1760kg/m^3(Mg) to 11400kg/m^3(Pb). Especially, ρ_p= 2520kg/m^3 (spherical glass beads) is selected for comparison with the experiment as shown later. The particle is thrown into the rotating tapered drum from the top of drum and at the point apart from the center of drum R_0=10mm.

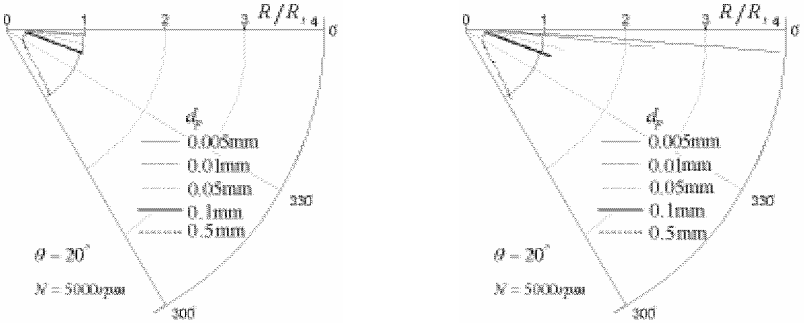


Fig. 2. Traces of particle motion, (a) V0=0m/s; (b) V0=1m/s

The initial vertical velocity of fluid V_0 plays an important role in separating the particle from the fluid from the viewpoint of the treatment performance of the TDRS.

Figure 2 (a) and (b) shows the relative traces of particle motion in the tapered drum using the particle diameter d_p as a parameter at the revolution of drum N=5000rpm in

the case of the initial vertical velocity V_0=0m/s and 1m/s, respectively. The most of length scales are normalized by using the top diameter of drum 78mm as a reference length, except the particle diameter. In these figures, the radial means the particle rotates with fluid (no relative motion). As the particle diameter becomes larger, the traces delay from the radial θ=0°gradually, because of the relative velocity between the particle and the fluid especially just at the beginning of motion. For example, at the beginning of particle motion thrown into the drum in the case of d_p=0.1mm, by means of the relative velocity, the particle moves to the rotational direction, whereas the smallest particle 0.005mm moves almost in radial direction. It is found from Fig. 2(b) that the reaching length in the case of especially very small particles 0.005mm and 0.01mm becomes long.

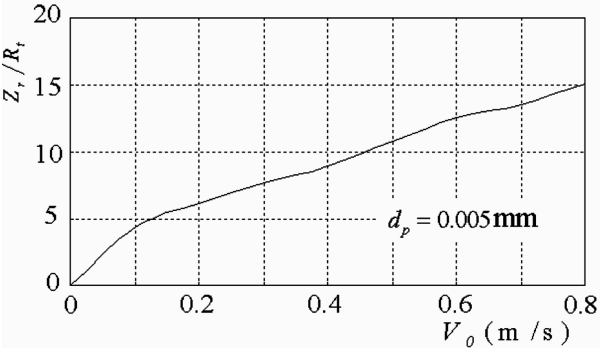


Fig. 3. Relation between vertical reaching length and initial vertical velocity

Fig.3 shows the relation between the vertical reaching length and the vertical velocity at d_p=0.005mm and N=5000rpm. From this figure, the drum length is to be determined considering the vertical velocity. For example, in the case of V_0=0.4m/s, the drum length is decided so Zr/Rt=9 as the particle to reach at the wall. But in this case, the cleaned water exit port is mounted at the center of the bottom of drum as shown in Fig.1, so the particle can be separated by using Zr/Rt less than 9. For V_0=0.8m/s, the length of drum is obtained as 15 from the same figure.

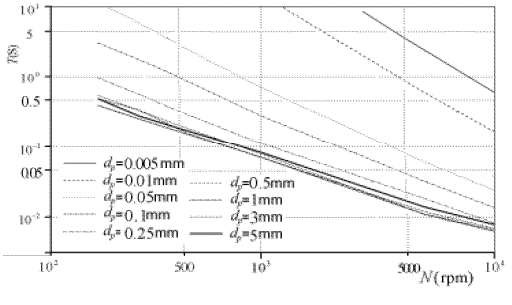


Fig. 4. Relations between reaching time and revolution of drum (Vo=0m/s)

The relations between the reaching time and the revolution of drum at V_0=0m/s are shown in Fig.4 using the particle diameter as a parameter. The reaching time decreases as the rotation of drum increases exponentially for each particle diameter. The reaching time shows the minimum value in the case of d_p=1mm (precisely speaking, d_p=1.2mm) depending on the balance of the centrifugal force and the gravity force acting upon a particle. As the particle diameter becomes larger than 1mm, the reaching time becomes longer again. So far as the reaching time is concerned, the particles larger than 1mm are not so much different compared with those smaller than 1mm.

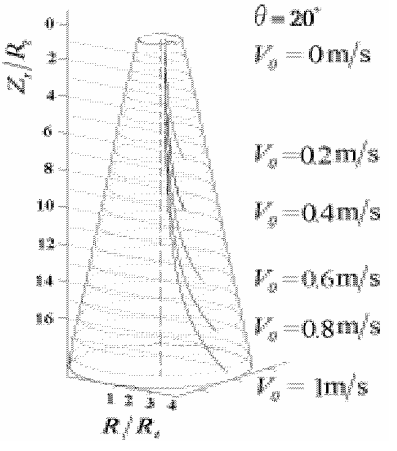


Fig. 5. Traces of particle for different vertical velocity (d_p=0.005mm)

Fig.5 shows the relative traces of the particle motion in the case of d_p=0.005mm using the vertical velocity as a parameter. The higher the initial vertical velocity, the longer the reaching length becomes. The particle moves with water in vertical direction and moves in radial direction against to the drag force. As shown in Fig.1, the cleaned water is extracted from the exit port at the center of the bottom of drum, so the particle is not necessary to reach at the wall. Therefore, it is expected to be separated even smaller particles in this condition.

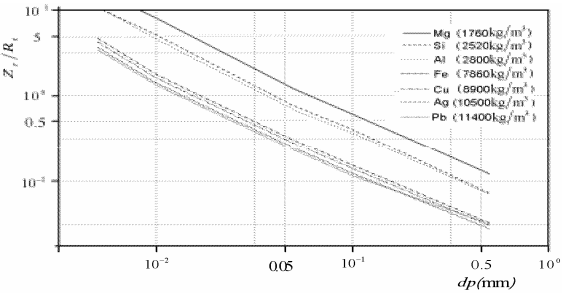


Fig. 6. Relations between vertical reaching length and density of particle (N=5000rpm, d_p=0.005mm)

Fig.6 shows the relations between the vertical reaching length Z_r and the diameter of particle d_p using the density of particle ρ_p as a parameter. As can be seen from this figure, the relations between Z_r and d_p is expressed approximately by

$$Z_r = \beta d_p^{-1/2} \tag{12}$$

where β is determined by the materials of particle.

4 Comparison with Experiments

In order to verify the calculated results, the experiments were carried out by using a flow visualization technique. The experimental apparatus and the instruments are shown in Fig.7. The spherical glass beads were used d_p=1mm and 2mm which are larger than those used in the numerical simulation, and the revolution of drum N=200rpm and 300rpm which are also lower than the practical ones, because of the difficulty of observation. The tapered drum is driven by the induction motor①, which is controlled by the inverter. The revolution of drum is detected by Digital tachometer. The particle is thrown from the particle feeding pipe② mounted the open/close valve into the Tapered drum③(transparent acrylic resin) filled with water. The particle motion is lightened by two Reflex lamps④ and taken picture by High-speed camera (PULNIX TM-6740GE)⑤. These data are recorded and analyzed by the Personal computer⑥.

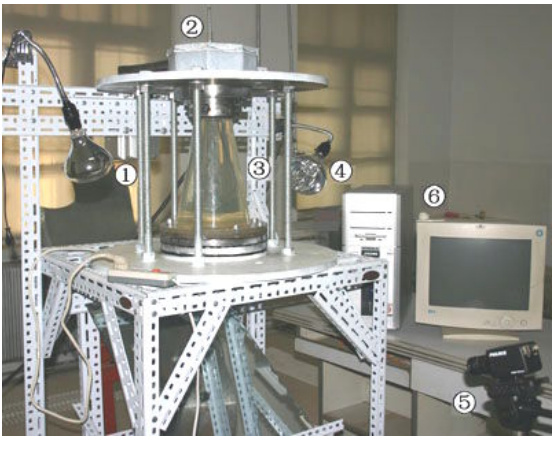


Fig. 7. Experimental apparatus and instruments

A time series of pictures were taken as 200frames/s. Some of these are shown in Fig.8 in the case of N=200rpm and d_p=2mm. These pictures show absolute traces by 0.1sec and 0.05sec time intervals for each. By analyzing these time series of pictures, Fig.9(a) and (b) shows the traces of particle motion in the case of d_p=1mm for N=200rpm and 300rpm, respectively, and Fig.10(a) and (b) in the case of d_p=2mm for N=200rpm and 300rpm, respectively. The calculated results are also shown in these figures for comparison. In the vicinity of the center region of the drum, the flow is

disturbed by the presence of the particle feeding pipe which does not move, so the flow field here is different from that used in the calculation model. However, except in this region, the calculated results of trace are in approximate agreement with experiment as a whole.

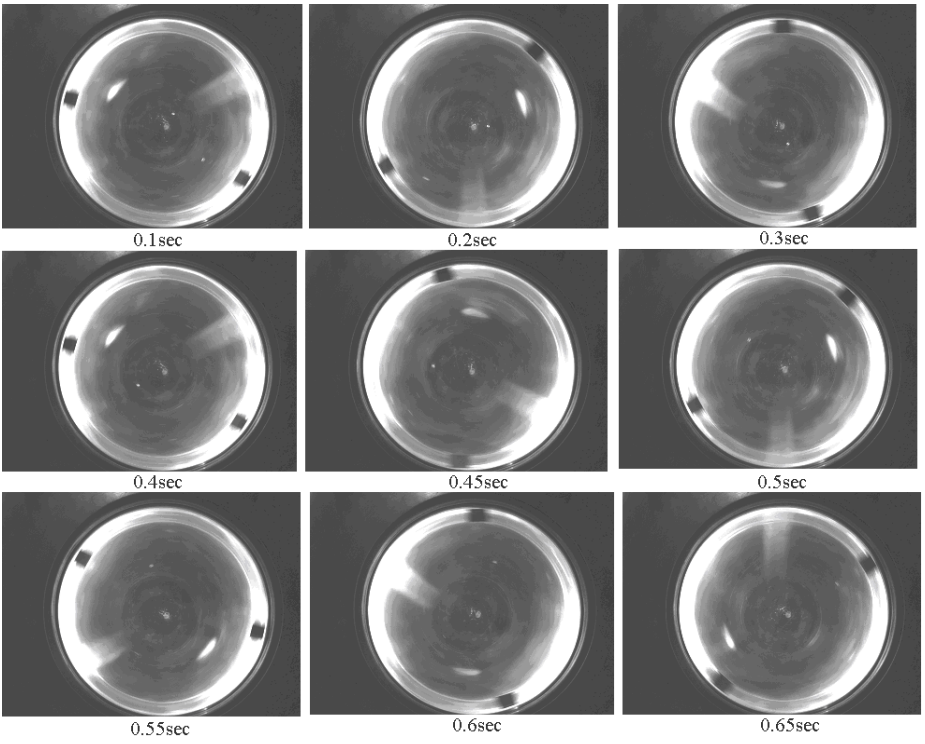
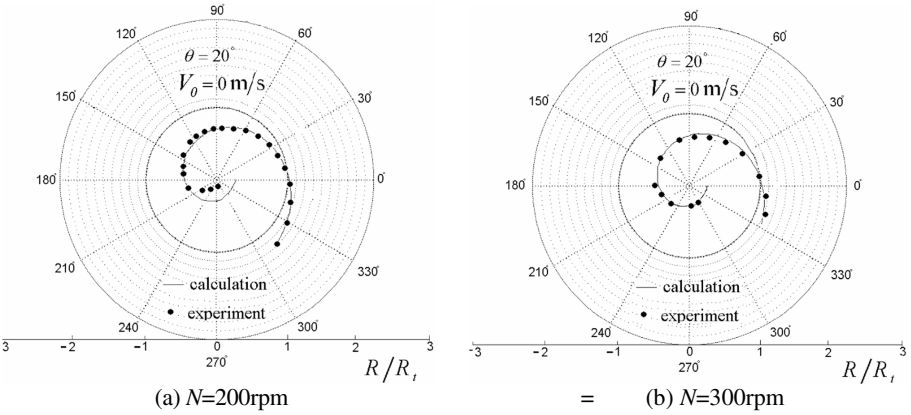
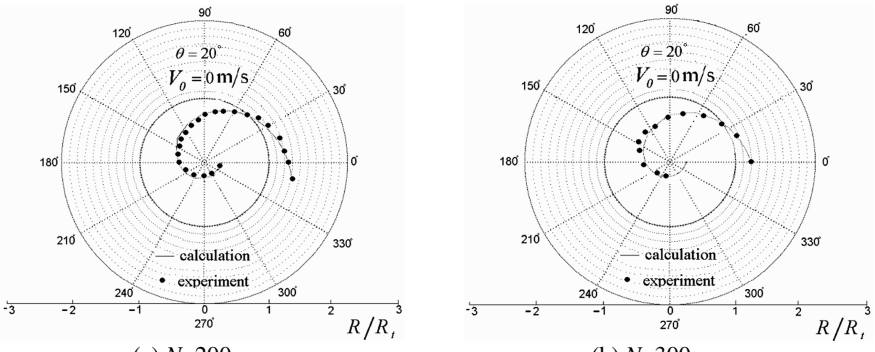


Fig. 8. A series of photographs of particle motion (N=200rpm, d_p=2mm and V_0=0m/s)



(a) N=200rpm = (b) N=300rpm

Fig. 9. Traces of particle motion in absolute coordinate (d_p=1mm)



(a) N=200rpm (b) N=300rpm

Fig. 10. Traces of particle motion in absolute coordinate (d_p=2mm)

5 Conclusions

As the fundamental study on a tapered drum rotating type separator (*TDRS*), the numerical simulation was carried out for relatively small particles and high revolution of drum. The top angle of drum and the vertical velocity were taken into consideration in the calculation of the particle motion in the rotating drum. At the same time, the experiments were carried out by using flow visualization technique. According to the numerical simulation and the experiment, it was clarified as follows:

(1) The length of drum should be determined by the density of treated particles and its smallest particle diameter and the revolution of drum, especially the particles smaller than 0.1mm being taken into consideration.

(2) The vertical reaching time at the wall of the particle is the shortest in the case of the particle diameter approximately 1mm by means of balance of the centrifugal force and the gravity force acting upon the particle.

(3) The particle whose density is smaller than 2000kg/m^3 is to be especially account for design of the drum.

(4) The calculated results of traces are in approximate agreement with experiment, except in the vicinity of the center region of the drum.

References

1. Zhu, T., Nozaki, T., Li, H.: Proc. of 9th International Symposium on Flow Visualization, Edinburgh, No. 411 (2000)
2. Zhu, T., Nozaki, T., Li, H.: Proc. of 1st Asian Particle Technology Symposium, Bangkok, No. 0097 (2000)
3. Zhu, T., Nozaki, T., Li, H., Fukuhara, M.: Fundamental Study of Tapered Drum Rotating Type Solid-Liquid Separator-Motion of Solid Particle Thrown into Rotating Cylindrical Container. Japanese Journal of Multiphase Flow 5(2), 435–440 (2001)
4. Saffman, P.G.: J.Fluid Mech. 22(2), 385 (1965)
5. Nozaki, T., Zhu, T., Li, H.: Proc. 2001 JSFM Symp (in Japanese), Tokyo, p. 645 (2001)

Impact on Genetic Algorithm of Different Parameters

Xiang Li, Qiao Chen, and Yanli Li

China University of Geosciences, No.388 Lumo Road Wuhan, P.R. China China
natural science fund No. 60473037
Lixiang@cug.edu.cn

Abstract. Genetic algorithm is a highly efficient parallel evolution algorithm of the global search which is decided by the balance of the exploitation and exploration. This balance also depends on population size, the biggest evolution of algebra, coding length, crossover rate and mutation rate, etc., therefore, the set of evolution parameters for the search properties of the entire GA has played a crucial role. This paper solved PERT in the multi-task scheduling problem, focused on population size, the cross-probability and the mutation rate of GA and analyzed their impact factors.

Keywords: Scheduling problems GA impact factor PERT.

1 Introduction

The realization of GA relates to the five elements: encoding parameters, the settings of initial group, the design of fitness function, genetic manipulation and the set of control parameters, and each element corresponds to the different environment and has corresponding design strategies and methods. Different strategies and means determine their own GA have different properties or characteristics. Therefore, the research of GA is very important. At present, GA evaluation is mainly used fitness value. There are many scholars abroad studying the impact of GA of the parameters set on, D. Whitey, in his thesis proposed Adjacency based crossover, applied it to the TSP problems, and proved it by experiments. H. Bersini and G. Seront combined GA and simplex method ,and formed a single operation called the simplex crossover and the results showed that the properties of three cross-operators is better than the other two. In China there are also many scholars who have improved the cross-GA operator. In 2007, Xiang Li, studied the use of GA in multi-task scheduling problem [1]. Huawei Yi, proposed the improved GA for Flow-shop scheduling problem[2]. Xiaolin Gu, gave improved GA and its application of other variations to the chaos Operators[3]. In 2008, Gui-Juan Chang, made the immune GA in the scheduling problem based on the orthogonal test[4]. Sun Jin Ling made GA based on the priority of the three individual cross-coding[5]. Each scholar has achieved certain results.

PERT in the multi-task scheduling problems is the typical of NP-hard problem[6], using the model to study the parameters of change on the impact of GA has representative and important significance. This paper focused on population size, cross-operator and the mutation operator of the impact of GA, and analyzed their impact factors.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 479–488, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Model of Multi-task Scheduling

Assuming there are several parallel tasks, and a shared resource pool containing a number of renewable resources, and all resources are supplied limitedly. In addition to sharing resources, tasks are independent with each other. To facilitate the description of the problem, the establishment of the mathematical model is as follows: RCPSP has P independent Multi-task scheduling tasks; the task k contains n_{k+1} works, the n_{k+1} task is terminating work for virtual tasks, and itself occupies no resources and time. P missions' works are to share this kind of M renewable resources; the total of the mth resource is Rm. W_i expresses the ith mission's work set, W_{ij} expresses in the ith task jth work, its duration is d_{ij}, the requirement of the mth resources is r_{ijm}, the starting time is S_{ij}, its all tight front duties formed the collection is P_{ij}. The set composed of all tasks on time t is named I_t. Considering the importance of different tasks, a_k is used for the weight k task. Taking these assumptions and symbols, a multi-task scheduling with resource constraints can be described as formula(1-6):

$$\min\sum_{k=1}^{P}\partial_k *(S_{k,n_k+1}) \tag{1}$$

$$s.t. \quad S_{i,j} \geq S_{i,h}+d_{i,h}, \quad \forall h \in P_{ij}, \forall i,j \tag{2}$$

$$\sum_{w_{i,j}\in I_t} r_{ijm} \leq R_m, \quad \forall m,t. \tag{3}$$

$$I(t)=\{Wkj \in \bigcup_{k=1}^{P} Wk \mid S_{kj} \leq t \leq S_{kj}+d_{kj}\} \tag{4}$$

$$r_{ijm} \geq 0 \tag{5}$$

$$\sum_{k=1}^{P}\partial_k = 1 \tag{6}$$

3 Algorithm Design

3.1 Data Structure for the Algorithm

Before the generation of initial population, it is first necessary to establish a data structure for storing information of the various tasks.

First, define a task structure.

```
struct project
{ int n;  int TP; int TC;   int  TS;    work  *w[MAX];
entity *E[MAX];}*pro;
```

Among them, n is the number of tasks, TP is the task completion time. TC is the calculated completion time of the task, TS is the actual completion time of the task, and work is the definition of the structure. w[MAX] task is working at the point-array.

Entity is the definition of the network structure. E[MAX] is the point-array related to the each task. Dynamic information to the importation of various tasks, Cross linked-list node is used store the relationship between the various tasks.

Cross linked-list node of the structure as follows:

```
typedef struct cnode
{ int hang; int lie; int value; struct cnode *down;
struct cnode *right;}crossnode;
typedef struct
{ int n; crossnode *hc[MAX];}clinknode; clinknode
*Head=new clinknode;
```

'hang' and 'lie' in the structure cross node mark the position in the linked-list. 'Value' is the index of the work, 'down' is the eligible point of it and 'right' is the immediate point. 'In the 'clink node' hc[MAX]' and 'Head' is the head note of cross linked-list.

In this paper, a key of the multi-task scheduling is setting-up a same cross linked-list for all of tasks. And the identification of various tasks is proposed in the linked-list. After the establishment of schedule, the activation of various tasks is based on the linked-list. This is also an innovation of this paper.

3.2 Chromosome Structure and Design Codes

Resource-constrained multi-task scheduling problems can be also considered as the scheduling problem with certain constraints. The key of solving the problem is how to use GA to find an appropriate sequencing of multitask. According to a certain sequence of chromosome, we code all of issues on the table. If there are m tasks, each task has n works, then there are m*n works. The first expression V_k is Kth chromosome in current population. Suppose P_{kij} is a gene of chromosome. It expresses a work of the P_kth task at location j of chromosome.

Chromosome can be expressed as follows:

$V_k=[P_{1i1},\ldots, P_{1ii},\ldots,P_{1in},\ldots,P_{ki1},\ldots, P_{kii},\ldots P_{kin},\ldots,P_{mi1},\ldots, P_{mii},\ldots,P_{min}]$

P_{kij} is bound by the random order with the constraint

3.3 Chromosomes Initialization

For the Chromosomes initialization, this paper was divided into two parts. One was randomly generating various chromosomes by the topological scheduling, the other was generating chromosome through a variety of heuristic methods.

Method 1. Randomly generated initial chromosome:

Similar with the optimal scheduling method with the shortest Resource Constrained time of the single task, the only difference is that for every work. Firstly judge the task to which it belongs, and then determine the constraint relationship. It is determined the first consider work by a Left to Right approach with a fixed sequence. At each stage of the work order has been maintained and assembled Scheduling can work and completely random from a pool selected to work in the pool. This process is repeated indefinitely until all of the work is arranged.

In each iteration of the process all the work at one of the following three conditions:

1. The work has index: Part of the chromosome structure in the work.
2. The work can schedule: All immediate works are ordered.
3. Free work: All of work

v_k^t is a part of chromosome, including t activities. Q_t is in phase t with a corresponding arrangements can be made to the activities set, Pi is all the activities set ,with a corresponding set of arrangements can be made activity set Q_t is defined as :

$$Q_t = \{ j \mid P_i \subset v_k^t \ \} \tag{7}$$

It includes specific arrangements in the end nodes, and it is all competition activity set.

Scheduling changes which may work in the following manner: a. deleted Qt from the work j which has been selected. b. judging if there is work k in all immediate works or not, and its all immediate works are ordered. c. if there is k, it can be added to the scheduling work set Qt.

Method 2. Initializing chromosome through various heuristic methods:

Bai Sijun listed 31 heuristic methods. Based on the resources 11 more effective heuristic ones are picked out here. Application of these 11 heuristic methods generates a good number of chromosomes of relatively activation effects.

The11 heuristic criteria are as follows:

(1) Minimum total time difference priority standard (MinTS), according to the earliest starting time if they are same.

(2) Maximum total time difference priority standard (MinTS), according to the earliest starting time if they are same.

(3) Minimum criteria for priority period (SOF), according to the numbers of work if they are same.

(4) Maximum criteria for priority period (SOF), according to the numbers of work if they are same.

(5) LS priority minimum standard (MinLS), according to the numbers of work if they are same.

(6) EF priority minimum standard (MinEF), according to the numbers of work if they are same.

(7) LF priority minimum standard (MinLF), according to the numbers of work if they are same.

(8) ACTIM standard, the different between the length of the critical path and the latest starting time, the greater the value has the priority.

(9) Maximum priority greatest resource requirements standard (MaxR).

(10) Minimum priority greatest resource requirements standard (MinR).

(11) Most intended tight works priority criterion (MostIS).

The literature[6] also makes some heuristic criteria considered: First considering the work on the critical path. If there are conflicts among several key tasks, considering the following programs:

(1)The smallest total time priority. (2)First start priority. (3) First completion priorities. (4) Priority for the most intense work, the remaining processing time and delivery time ratio is smaller becomes tenser.

3.4 Genetic Operators Design

On the base of established cross linked-list, re-number various tasks in the work. Then apply the algorithm described in this paper. All tasks will be mixed with the work. After encoding into a chromosome, the problems become simple. Genetic chromosome operation with the single task is similar to the chromosomal genetic operations.

3.4.1 Crossover

PMX is used in literature[7] (Partly Matched Crossover), but the operator would have to generate) illegal entity. And it will need to repair chromosome, which will reduce the efficiency of the implementation process. Here it proposes a method that it is not used to repair chromosome according to the discrete crossover operator.

Participate in the two crossover operator for mothers M and fathers F. We choose a random integer q,1<=q<=J,J is chromosome length. M and F through radio spots overlapping operations produced two generations of daughters D and sons S. In the list D work, the former q tasks come from M.

$j_i^D = j_i^M$,i = 1, 2…q

i = q + 1,…J, come from the location of the F, And the relative position of the F resurveyed.

$j_i^D = j_k^F$, i = q +1, q + 2,…,J

Which $k = \min\{k \mid j_k^F \notin \{j_1^D, j_2^D, \cdots, j_{i-1}^D\}, k = 1,2,\cdots, J\}$

S and D are similar to the formation of the linked-list, not going to repeat here.

3.4.2 Mutation

Mutation operator adopts centralized search strategy combined with the neighborhood technology to improve the offspring. D genes will not exceed the change in the scheduling of the neighborhood gathered as scheduling x (Fig. 1). If x neighborhood than any other solution, then x is called scheduling optimization. Mutation process is as follows:

```
  Begin
  If (rand () <pmutation)
  Take n continuous in a row as the chromosome genes
  Permutations and combinations of the n genes
  Check each gene sequence, the genome sequence will be its
discarded if it unreasonable;
  Assessing all the neighborhood scheduling;
  Neighborhood as the best choice for future generations
  End
```

Individual i

A	B	C	D	F	E	H	G	I

The neighborhood of individual i

A	B	C	F	D	E	H	G	I
A	B	D	C	F	E	H	G	I
A	B	D	F	C	E	H	G	I
A	B	F	C	D	E	H	G	I
A	B	F	D	C	E	H	G	I

Fig. 1. Neighborhood maps chromosome

In the process of mutation, two consecutive complete gene permutations and combinations. Of course, there has a violation of constrain sequence. Here can directly be discarded, but will not have an impact on the choice of the best neighborhood. This method avoids repairing the chromosome after mutation and improves the efficiency of the procedure.

3.5 Selection Operator

The operation that winning individuals are chosen from the group and the bad individuals are eliminated is named selection.

Selection is based on the currently popular " Breeding Pool choice", "To adapt to-value ratio of options", " Ranking Selection ","Mechanism based on local competitive choices", and so on. Roulette Wheel Selection [8] is based on fitness than A choice of the most widely used methods. It is the first calculation of the relative fitness of individual $f_i/\sum f_i$, named P_i, and then chooses probability $\{P_{i,} = 1,2,\ldots, N)$ a disk cut into N copies, $2\pi Pi$ fan angle of the center of Pi. In making the selection, rotating disk can be assumed that if a reference point to fall into the itch fan, then chose individual i. Generation a random number r in [0, 1], if $P_o+P_i+\ldots +P_{i-1}$ "$r<P_l+P_2\ldots +P_i$, i individual choice, the assumption here $P_o=0$. It is easy to see that this was very similar to roulette choice of trouble. For sector bigger area, it has the greater the probability fall in and it was an opportunity to be chosen.

Thus, the structure of the gene was likely to pass it on to the next generation greater. This paper discussed the use of multi-robin scheduling optimization algorithm choice. Every generation of new groups in the use of optimal preservation strategy that will preserve the best, so far in the contemporary individual and overcome random sampling error.

3.6 Fitness Function

Calculate the shortest duration of tasks, we must first decode chromosome. Calculate the earliest starting time, and then find the earliest completion time for each task. Finally, according to the weight ratio of the various tasks calculate the shortest weighted average duration. Reached the objective function formula (8):

$$\min\sum_{k=1}^{P}\partial_k *(S_{k,n_k}+1) \tag{8}$$

Since the task is to minimize the total built-constrained Project Scheduling Problem issues. We change the original objective function to ensure that the individual is suited to meet the greatest value. The current population v_k is located k chromosome; g (v_k) is a fitness function. f (v_k) is the original target value. f_{max} and f_{min} are the biggest target value and the minimum target value. Conversion methods such as formula (9) :

$$g(v_k) = \frac{f_{\max} - f(v_k) + \varepsilon}{f_{\max} - f_{\min} + \varepsilon} \tag{9}$$

Where ε is a positive number, often limited the open interval (0,1).

4 Applications and Analysis

4.1 Examples and Data

The examples of cases come from literature [8]. This is a task of three parallel scheduling problems with the same tasks of the three network structure. The network structure is shown in Fig. 2. Each task has 16 works, all 48 work share 9 resources. The resources are here alone mode. Every of them can only be used as a resource. Different tasks have different durations, and types of resource requirements and demand, as shown in Table 1. The type and total of resources is as shown in Table. 2.

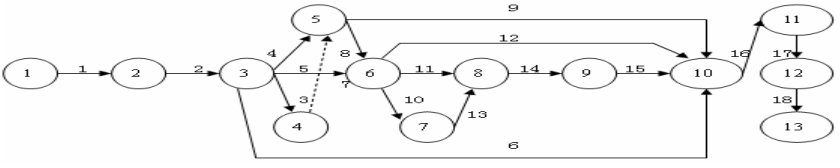


Fig. 2. Two mission codenamed network structure

From the results we can see after using GA the fitness of illumination increases, in a word, all trees can obtain more and more huge quality of illumination. Comparing Figure 3 with Figure 4 we can find the forest-environment model established by GA can preferably show up the space character of tree growing, sufficiently reveal the phototropism of plants and the effect between plants and recur the growing process realistically.

4.2 The Impact Analysis of Population Factor

For a given item weight coefficient a1 = 0.3, a2 = 0.24, a3 = 0.46, Population size is 20 to 200, the biggest evolutionary times is 100, the crossover probability is 0.8, the mutation rate is 0.01, and the mutation in the neighborhood length (λ) is set to 4. Set up three projects started the same time. When the evolutionary time is 57 and the population size is 120, the fitness function convergence, and this fitness function value is 75.08. Obviously, this example for the size of the population is not bigger the better, and the impact of its objective functions is showed in Fig. 3.

Table 1. Task the resources required type and duration of work demand

Work Index	Begin Pot	Resources type	Mission 1		Mission 2		Mission 3	
			Resources	time	Resources	time	Resources	time
1	1,2	1	6	4	8	5	10	4
2	2,3	1	8	3	7	6	8	6
3	3,4	4	4	2	5	2	4	1
4	3,5	1	10	4	9	3	7	4
5	3,6	3	2	3	3	2	4	3
6	3,10	2	3	2	4	6	5	6
7	4,5	1	0	0	0	0	0	0
8	5,6	1	5	4	6	6	9	3
9	5,10	2	4	1	4	3	3	2
10	6,7	7	10	2	14	1	11	1
11	6,8	6	9	4	11	2	8	2
12	6,10	7	12	3	9	3	10	3
13	7,8	5	7	4	5	4	7	3
14	8,9	6	13	3	10	2	12	4
15	9,10	7	16	2	13	4	14	2
16	10,11	8	6	3	5	3	16	4
17	11,12	9	3	4	4	4	5	4
18	12,13	1	0	0	0	0	0	0

Table 2. Types and total of the resources

Number	Name	Total
1	Designer	10
2	Purchaser	6
3	Processor	4
4	Programmer	5
5	Heat treatment center	8
6	Electric machining center	14
7	Nc machining center	18
8	Assembly center	7
9	Number module center	5

4.3 The Impact Analysis of Cross-Operator

For a given item weight coefficient a1 = 0.3, a2 = 0.24, a3 = 0.46, Population size is 50, the biggest evolutionary times is 100, the crossover probability is 0.1 to 1.0, the mutation rate is 0.01, and the mutation in the neighborhood length (λ) is set to 4. Set up three projects started the same time. When the evolutionary time is 48 and the crossover probability is 0.35, the fitness function convergence, and this fitness function value is 82.36. Obviously, this example for the size of the population is not bigger the better, and the impact of its objective functions is showed in Fig. 4.

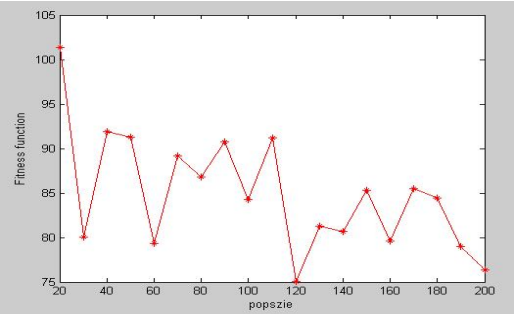


Fig. 3. The impact result of population size

4.4 The Impact Analysis of Mutation-Operator

For a given item weight coefficient a1 = 0.3, a2 = 0.24, a3 = 0.36, Population size is 50, the biggest evolutionary times is 100, the crossover probability is 0.8, the mutation rate is 0.01 to 1.0, and the mutation in the neighborhood length (λ) is set to 4. Set up three projects started the same time. When the evolutionary time is 63 and the crossover probability is 0.1, the fitness function convergence, and this fitness function value is 83.24. Obviously, this example for the size of the population is not bigger the better, and the impact of its objective functions is showed in Fig. 5.

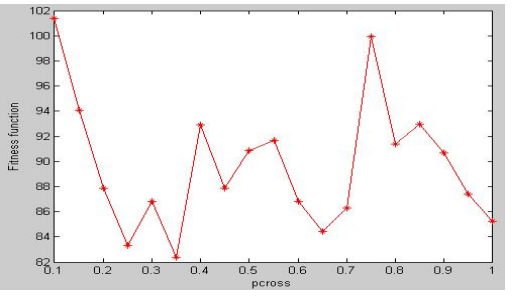


Fig. 4. The impact result of cross-probability

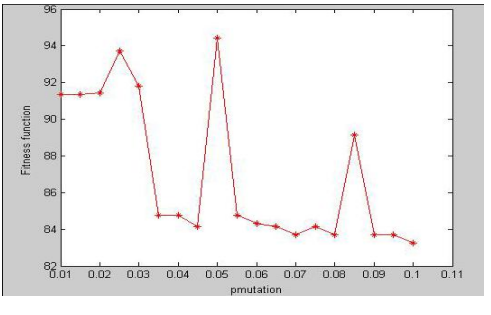


Fig. 5. The impact result of Mutation probability

5 Conclusion

This paper discussed the impact of using the population size, the cross-probability and mutation rate on GA fitness goals that so far has not recognized a kind of rip principles and methods of the setting parameters. The set of parameters has often been from experience and prior knowledge of the problem. However, due to the process of evolution of GA continuously promoted, the average fitness, and many other natures are constantly changing, so a fixed algorithm parameter is difficult to apply to the entire course of evolution, it is proposed of adaptive GA for dynamic set of parameters.

References

1. Li, X., Kang, L., Tan, W.: Optimized Research of Resource Constrained Project Scheduling Problem Based onGenetic Algorithm. In: Kang, L., Liu, Y., Zeng, S. (eds.) ISICA 2007. LNCS, vol. 4683, pp. 177–186. Springer, Heidelberg (2007)
2. Yin, H., Zhang, Q.: Improved genetic algorithm for permutation Flow-shop scheduling problem. Computer Engineering and Application 22 (2007)
3. Gu, X., Huang, M., Liang, X.: Improved genetic algorithm based on chaotic mutation operation and its application. Computer Application 10 (2007)
4. Chang, G., Zhang, J.: Immune Genetic Algorithm Based on Orthogonal Experiment for Scheduling Problems. Information and Control (2008)
5. Sun, J., Chang, G., Zhang, H., Xu, Y.: A Genetic Algorithm Based on Three-Individual-Crossover and Preference Weight Coding. Journal of Qingdao Agricultural University (2008)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness[M]. Freeman, San Francisco (2006)
7. Xuan, G., Cheng, Z.: Genetic Algorithm and Engineering Design. Science publishing company (2000)
8. Feng, C.-W., Liu, L., Burns, S.A.: Using Genetic Algorithm to Solve Construction Time-Cost Trade-Of Problems. Journal of Computing in Civil Engineering 11(3), 184–189 (2006)

New Fast Decision Tree Classifier for Identifying Protein Coding Regions

Hazem M. El-Bakry[1] and Mohamed Hamada[2]

[1] Faculty of Computer Science & Information Systems,
Mansoura University, Egypt
helbakry20@yahoo.com
[2] University of Aizu
Aizu Wakamatsu, Japan
Hamada@u-aizu.ac.jp

Abstract. In this paper, a fast tool for finding protein coding regions is presented. Such tool relies on performing cross correlation in the frequency domain and decision Tree. In addition, a modified trust region method is used to find the closet (optimized) DNA nucleotide. Moreover, a Sequential PRM-based protein folding algorithm for finding the point where these proteins add to the ladder is introduced. Furthermore, standard parallel scan algorithm is used to provide parallel processing of the strides and its transitions. This proposed tool produces more accurate results, than that have previously been obtained for a range of different sequence lengths. Experimental results confirm the scalability of the proposed classifying tool to handle large volume of datasets irrespective of the number of classes, tuples and attributes. High classification accuracy is achieved. The main achievement in this paper is the fast decision tree algorithm. Such algorithm relies on performing cross correlation in the frequency domain between the input data at each node and the input weights of neural networks. It is proved mathematically and practically that the number of computation steps required for the presented FNNs is less than that needed by conventional neural networks (CNNs). Simulation results using MATLAB confirm the theoretical computations.

1 Introduction

Genes carry the instructions for making proteins that are found in a cell as a specific sequence of nucleotides which are found in DNA molecules. But, the regions of these genes that represent code for proteins may occupy only a small region of the sequence. Identifying the coding regions plays a vital role in understanding these genes [1,2]. Many of the challenges in biology are now challenges in computing. Bioinformatics [4], the application of computational techniques to analyze the information associated with bimolecules on a large scale, has now firmly established itself as a discipline in molecular biology. Bioinformatics is a management information system for molecular biology. Bioinformatics encompasses everything from data storage and retrieval to the identification and presentation of features within data, such as finding

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 489–500, 2008.
© Springer-Verlag Berlin Heidelberg 2008

genes within DNA sequence, finding similarities between sequences, structural predictions [5]. Analyzing the coding regions is not the scope of this research. Finding the coding regions is difficult because, Genomes are small are not continuous (0.1-10.1bp). The coding density [10] in eukaryotes is less than 50% till date (Only 5% of genes constitutes of coding regions). DNA is arranged in the form of exons and introns as shown in Fig. 1. Introns form maximum part of DNA and extons form the minor part of DNA. But these extons contain the small protein coding regions.

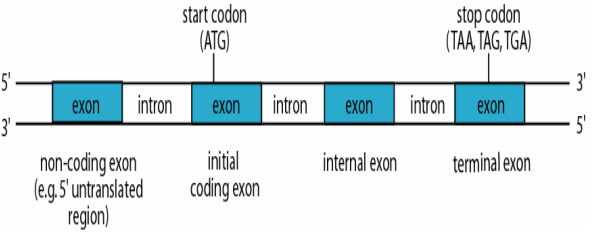


Fig. 1. DNA Sequence in form of strides of three

But, Proteins attach to the DNA and help the strands coil up into a chromosome when the cell gets ready to divide. The point where the proteins add to the strands is very important to this research. This tool uses a Sequential PRM-based protein folding algorithm [5] for finding the point where these proteins add to the ladder. We also need to find the closet neighbor hood stride, which is the main reason for the cause of dynamism in this research project. We modify trust region method for finding the optimized stride [6]. Once we find the neighboring strides we use parallel scan algorithm for processing the sequence to find the coding region. The concept of induction of decision trees is described in the section 2. In section 3, a modified trust-region method is discussed. Section 4 covers the parallel scan algorithm and section 5 discusses protein folding algorithm. Section 6 gives details regarding data and discusses briefly on the experimental results. Our new contribution, fast detection of protein coding regions by using neural networks and cross correlation in the frequency domain is presented in section 7.

2 Induction of Decision Trees

Decision tree algorithms have been described in many places. The basic algorithm will only be summarized here. All decision tree systems operate using the same underlying algorithm. The input is a set S of examples; in the current study, S consists of a set of non overlapping DNA subsequences.

2.1 Identification of Protein Coding Regions in DNA Sequence

The algorithm is:

Input: DNA Sequence
Output: Decision Tree

1. Split the set S using a test on one or more features, f1, f2, ..., fd.(Using Parallel Scan Algorithm, Section 4)
2. Check the results of the split. If every partition is pure (all examples in the Partition belong to the same class), then stop. Label each leaf node with the name of the class there.(Using the PRM-based protein folding algorithm, Section 5)
3. Recursively Calculate the nearest neighbor (Optimized neighbor using Trust Region Method, Section 3)
4. Recursively split any partitions that are not pure.

The result of this algorithm is a decision tree with test nodes and leaf nodes. The leaf nodes contain class labels. Decision trees vary primarily in how they choose a ÒsplitÓ of the data, and in how they prune the trees they produce. The data should be formatted as a set of features and a class label; for example, this study includes 21 coding measures (the features), such as dicodon frequencies, hexamer frequencies, and length of the longest open reading frame. Each of these features is a number computed on a fixed-length window of the DNA sequence[1,2]. There are only two class labels: coding and noncoding.

2.2 Splitting Rules

Given this numeric input, a decision tree algorithm must decide how to use the features to ÒsplitÓ the data. For example, if a feature is named Òcodon usage,Ó one possible test might be codon usage < 2.45. We call this a univariate test because it only uses one feature (variable). With a univariate test, there are only N-1 distinct tests to consider for N examples. These would occur at the midpoints between successive feature values. With N examples and D features, we have D(N-1) such tests to consider. Each test is scored by a goodness measure. The simplest such measure is the one that counts the number of features that would be misclassified by the test (Heath et al. 1993). More sophisticated tests measure the entropy of the initial set and the subsets produced after splitting (Quinlan 1993), or use statistical properties such as the probabilities of each class on either side of the split. (Note that a test always splits the data into two subsets.) The Òtwoing ruleÓ (Breiman et al. 1984) tries to create tests that maximize the purity of both subsets while at the same time splitting the data roughly in half. For definitions of these and other goodness measures, see Murthy et al. (1994). Another, more complicated type of test is the linear discriminant, which was used by Fickett and Tung (1992). In the context of decision trees, this kind of test is called an *oblique* split. Instead of testing a single feature such as codon usage, these tests include a multivariate combination of the features f1, f2, ..., fd, in the form: a1f1 + a2f2 + ... + adfd ³… ad+1. This test, while much more powerful than the simpler univariate test, is also more expensive to compute. However, recent work on randomized decision tree algorithms has resulted in very efficient methods for finding these oblique tests (Murthy et al. 1994). Our THE PROPOSED TOOL system always finds the best axis-parallel split first, and then considers oblique splits and uses those if they work better. The system can be set to use axis-parallel splits only for even greater efficiency. One interesting result that came out of the current study was that virtually all goodness measures are designed so as to maximize, in one form or another, the overall accuracy of the tree. Thus if one class is much larger than another, as is the

case with Fickett and TungÕs benchmark training sets, decision trees tend to optimize accuracy on the larger class. To correct this, a new goodness measure called Òaverage accuracy was developed [1,2]. Average accuracy measures the goodness of a split by simply computing the accuracy of each class for a given test, and then averaging the results across all classes. In order to use all the other goodness measures, though, training sets in which the sizes of the classes were roughly equal are produced. From each of the benchmark data sets, therefore, a subset of the training data with equal class distributions, and used just the subset for building trees were extracted.

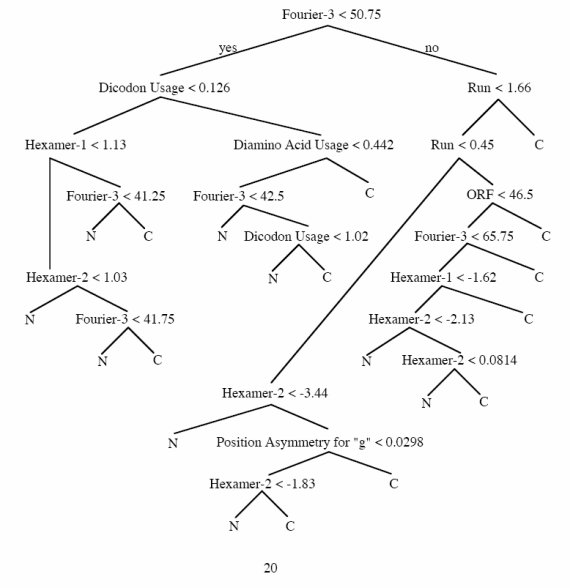


Fig. 2. Classification of DNA sequence with several measures

2.3 Pruning Rules

Most decision tree systems provide some means for pruning the decision tree after it is built. The initial algorithm takes some training data and builds a complete tree, one that correctly classifies every example in the training set. When the data is noisy, this tree usually ÒoverfitsÓ the training data and performs poorly on additional data. A smaller tree often performs better, and offers the advantage of being simpler and faster to use. Our decision tree system uses a pruning technique called cost complexity pruning. First it divides the training data into two sets, the training set (T) and the pruning set (P). After building a decision tree on T, it then produces a succession of smaller trees by looking at every non-leaf node in the tree, and measuring the cost complexity of that node. Cost complexity considers the number of examples that would be misclassified (the cost) if that node were made into a leaf, and it also measures the complexity of the subtree rooted at that node. These two are combined into the cost complexity measure. The node whose cost complexity is greatest is then deleted or Òpruned,Ó and the process continues. This continues until the tree is pruned down to a single node. The system

then examines the series of increasingly smaller trees and computes their accuracies on the pruning set P. In the simplest pruning model, called SE-0 pruning, it chooses the tree with the highest accuracy on the pruning set. The SE-1 pruning method computes the standard error (SE) of the accuracies of all the trees and then chooses the smallest tree whose accuracy is within one standard error of maximal. In this paper, SE-0, SE-0.5, and SE-1 pruning are considered.

3 A Modified Trust-Region Method

Here, a modified trust-region method for unconstrained optimization is introduced. The radius of the corresponding neighbors DNS stride update is computed by using the model information at the current iterate rather than at the preceding one. The update is then performed according to how well the current model retrospectively predicts the value of the objective function at the previous iterate. Global convergence torts- and second-order critical point is proved under classical assumptions and preliminary numerical experiments on cuter problems indicate that the modified method is very competitive. Trust-region methods are well-known techniques in nonlinear no convex programming, whose concept has matured over more than thirty years. In such methods, one considers a model mk of the objective function which is assumed to be adequate in a _trust region_, which is a neighbourhood of the current iterate x_k. This neighbourhood is often represented by a ball in some norm, whose radius k is then updated from iteration k to iteration k + 1 by considering how well m_k predicts the objective function value at iterate x_{k+1}. In retrospect, this might seem unnatural since the new radius _k+1 will determine the region in which a possibly updated $modelm_{k+1}$ is expected to predict the value of the objective function around x_{k+1}. Our aim in this paper is to propose a variant of the trust-region algorithm that determines _k+1 according to how well mk+1 predicts the value of the objective function at xk, thereby synchronizing the radius update with the change in models. This paper uses the theoretical properties and practical numerical potential of the new trust region algorithm.

4 Fast Scan Algorithm

An important primitive for (data) parallel computing is the scan operation, also called prefix sum which takes an associated binary operator and an ordered set [a1,...,an] of n elements and returns the ordered set example, plus scan([1,0,3,4,5,6,7]) = [1,3,7,0,4,1,6,3]. Fig 3,4 show all the possible transitions of the strides from one to four. Notice that computing the scan of an n-element array requires n − 1 serial operations. Assuming we have n DNA sequence, each with one element of the array.

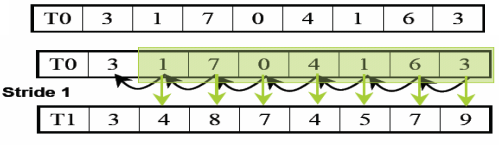


Fig. 3. Example Stride transitions

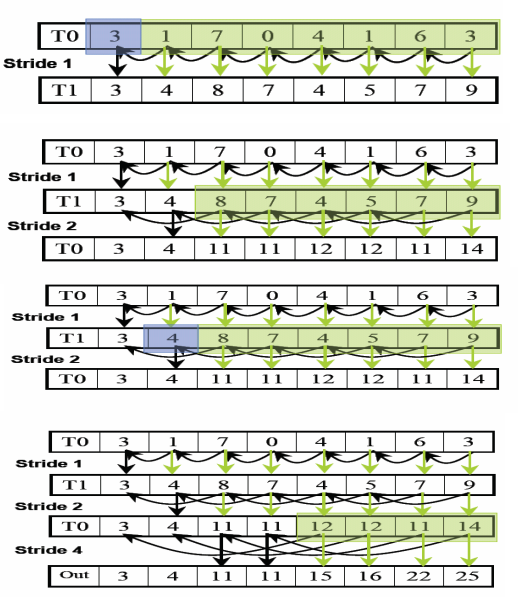


Fig. 4. Example Stride transitions

5 Protein Folding Algorithm

The DNA is organized into stretches of genes, stretches where proteins attach to coil the DNA into chromosomes, stretches that "turn a gene on" and "turn a gene off", and large stretches whose purpose is not yet known to scientists. Proteins attach to the DNA and help the strands coil up into a chromosome when the cell gets ready to divide. The genes carry the instructions for making all the thousands of proteins that are found in a cell. The proteins in a cell determine what that cell will look like and what jobs that cell will do. PRM-based protein folding algorithm for finding the point where these proteins add to the ladder. In contrast to most current folding algorithms, it uses very few energy parameters. Remaining conformational space of a simplified real-space representation of chains to find a minimum energy of an exceedingly simple potential function. This algorithm returns 'r' which is the point where these proteins attach DNA strand.

Input: Protein's native state q_{native}
Output: A roadmap R of the protein's potential energy landscape
Generate n nodes biased towards q_{native}
for each $q \in R$ do
    Let $N_k(q)$ be the k closest neighbors of q
    for $q' \in N_k(q)$ do
        if the local planner can find a path between q and q' then
            add edge (q, q') to R
        end if
    end for
end for
return r

6 Experimental Results

The data used for this study are the human DNA data collected by Fickett and Tung. All the sequences are taken from GenBank in May 1992. Fickett and Tung have provided the 21 different coding measures that they surveyed and compared. The benchmark human data include three different datasets. For the first dataset, non-overlapping human DNA sequences of length 54 have been extracted from all human sequences, with shorter pieces at the ends discarded. Every sequence is labeled according to whether it is entirely coding, entirely non-coding, or mixed, and the mixed sequences (i.e., overlapping the exon-intron boundaries) are discarded. The dataset also includes the reverse complement of every sequence. This means that one-half of the data is guaranteed to be from the non-sense strand of the DNA. In the next section we will give the experimental results for finding this coding region for all sequence lengths. Tables 1, 2, and 3 show the predictive accuracy of different algorithms on both coding and non-coding DNA sequences. In this section we present the results of THE PROPOSED TOOL classifier for Fickett and Tung's dataset. Values are given for the percentage accuracy on test set coding sequences and the percentage accuracy on test set non coding sequences.

Table 1. Predictive Accuracy for length 54 human DNA Sequence

Algorithm	Coding	Non Coding
Dicodon Usage	61%	57%
Bayesian	51%	46%
SUCA	78%	72%
UN FMACA	79%	72%
The proposed tool	79.5%	72.8%

Table 2. Predictive Accuracy for length 108 human DNA sequence

Algorithm	Coding	Non Coding
Dicodon Usage	58%	50%
Bayesian	45%	36%
SUCA	74%	69%
UN FMACA	75%	70%
The proposed tool	76%	73%

Table 3. Predictive Accuracy for length 252 human DNA sequence

Algorithm	Coding	Non Coding
Dicodon Usage	65%	54%
Bayesian	50%	44%
SUCA	71%	70%
UN FMACA	73%	72%
The proposed tool	74.5%	76%

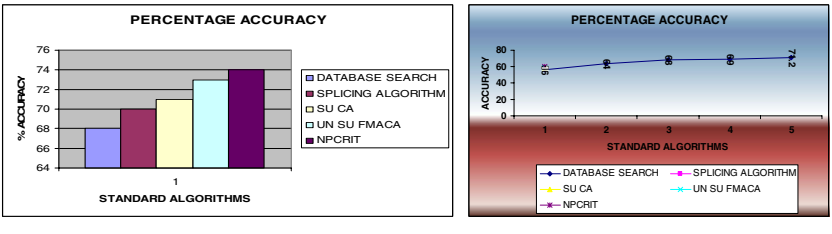


Fig. 5. Percentage Accuracy for 162

The drawback of Database Search is it fixes the gene and predicts the protein coding regions. Splicing algorithm only focuses on the length 108 DNA sequence. The Supervised CA is best for predicting 54 and 108 length DNA sequences regions. The UN supervised FMACA is used for predicting regions with DNA codon measure less than 56bp. But the proposed tool is used to find coding regions for all sequence lengths, and it accommodates dynamism by not fixing the gene at the attachment of proteins. The percentage accuracy reported was 75%. Fig. 5 (Data base Search, Splicing Algorithm, Supervised CA, Un Supervised FMACA, THE PROPOSED TOOL) shows that THE PROPOSED TOOL can be used to identify protein coding regions and the results obtained were comparable with other standard algorithms. The proposed tool overcomes all the disadvantages of previous standard algorithms like fixing the position of the gene and static order of the DNA sequence. The average accuracy reported is 75%.

7 Fast Detection of Protein Coding Regions by Using Neural Networks and Cross Correlation in the Frequency Domain

Finding a certain pattern of information, in the incoming serial data, is a searching problem. First neural networks are trained to classify the required pattern from other examples and this is done in time domain. In pattern detection phase, each position in the incoming matrix is tested for presence or absence of the required pattern. At each position in the input one dimensional matrix, each sub-matrix is multiplied by a window of weights, which has the same size as the sub-matrix. The outputs of neurons in the hidden layer are multiplied by the weights of the output layer. When the final output is high, this means that the sub-matrix under test contains the required pattern and vice versa. Thus, we may conclude that this searching problem is a cross correlation between the incoming serial data and the weights of neurons in the hidden layer.

The convolution theorem in mathematical analysis says that a convolution of f with h is identical to the result of the following steps: let F and H be the results of the Fourier Transformation of f and h in the frequency domain. Multiply F and H* in the frequency domain point by point and then transform this product into the spatial domain via the inverse Fourier Transform. As a result, these cross correlations can be represented by a product in the frequency domain. Thus, by using cross correlation in the frequency domain, speed up in an order of magnitude can be achieved during the detection process [14-20]. Assume that the size of the protein coding region is 1xn. In protein coding region detection phase, a sub matrix I of size 1xn (sliding window) is extracted from the tested matrix, which has a size of 1xN. Such sub matrix, which may contain the required code, is fed to the neural network. Let W_i be the matrix of weights between the input sub-matrix and the hidden layer. This vector has a size of 1xn and can be represented as 1xn matrix. The output of hidden neurons h(i) can be calculated as follows:

$$h_i = g\left(\sum_{k=1}^{n} W_i(k)I(k) + b_i\right) \tag{1}$$

where g is the activation function and b(i) is the bias of each hidden neuron (i). Equation 1 represents the output of each hidden neuron for a particular sub-matrix I. It can be obtained to the whole input matrix Z as follows:

$$h_i(u)=g\left(\sum_{k=-n/2}^{n/2} W_i(k)\, Z(u+k)+b_i\right) \tag{2}$$

Equation 2 represents a cross correlation operation. Given any two functions f and d, their cross correlation can be obtained by:

$$d(x)\otimes f(x)=\left(\sum_{n=-\infty}^{\infty} f(x+n)d(n)\right) \tag{3}$$

Therefore, Eq. 2 may be written as follows [20]:

$$h_i = g\left(W_i \otimes Z + b_i\right) \tag{4}$$

where h_i is the output of the hidden neuron (i) and $h_i(u)$ is the activity of the hidden unit (i) when the sliding window is located at position (u) and (u) $\in$ [N-n+1].

Now, the above cross correlation can be expressed in terms of one dimensional Fast Fourier Transform as follows [20]:

$$W_i \otimes Z = F^{-1}\left(F(Z)\bullet F^{*}\left(W_i\right)\right) \tag{5}$$

Hence, by evaluating this cross correlation, a speed up ratio can be obtained comparable to conventional neural networks. Also, the final output of the neural network can be evaluated as follows:

$$O(u)=g\left(\sum_{i=1}^{q} W_O(i)\, h_i(u)+b_O\right) \tag{6}$$

where q is the number of neurons in the hidden layer. O(u) is the output of the neural network when the sliding window located at the position (u) in the input matrix Z. W_o is the weight matrix between hidden and output layer.

The complexity of cross correlation in the frequency domain can be analyzed as follows:

1- For a tested matrix of 1xN elements, the 1D-FFT requires a number equal to $N\log_2 N$ of complex computation steps [21]. Also, the same number of complex computation steps is required for computing the 1D-FFT of the weight matrix at each neuron in the hidden layer.

2- At each neuron in the hidden layer, the inverse 1D-FFT is computed. Therefore, q backward and (1+q) forward transforms have to be computed. Therefore, for a given matrix under test, the total number of operations required to compute the 1D-FFT is $(2q+1)N\log_2 N$.

3- The number of computation steps required by FNNs is complex and must be converted into a real version. It is known that, the one dimensional Fast Fourier Transform requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions [13]. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. Therefore, the total number of computation steps required to obtain the 1D-FFT of a 1xN matrix is:

$$\rho=6((N/2)\log_2 N) + 2(N\log_2 N) \tag{7}$$

which may be simplified to:

$$\rho=5N\log_2 N \tag{8}$$

4- Both the input and the weight matrices should be dot multiplied in the frequency domain. Thus, a number of complex computation steps equal to qN should be considered. This means 6qN real operations will be added to the number of computation steps required by FNNs.

5- In order to perform cross correlation in the frequency domain, the weight matrix must be extended to have the same size as the input matrix. So, a number of zeros = (N-n) must be added to the weight matrix. This requires a total real number of computation steps = q(N-n) for all neurons. Moreover, after computing the FFT for the weight matrix, the conjugate of this matrix must be obtained. As a result, a real number of computation steps = qN should be added in order to obtain the conjugate of the weight matrix for all neurons. Also, a number of real computation steps equal to N is required to create butterflies complex numbers ($e^{-jk(2\Pi n/N)}$), where 0<K<L. These (N/2) complex numbers are multiplied by the elements of the input matrix or by previous complex numbers during the computation of FFT. To create a complex number requires two real floating point operations. Thus, the total number of computation steps required for FNNs becomes:

$$\sigma=(2q+1)(5N\log_2 N) +6qN+q(N-n)+qN+N \tag{9}$$

which can be reformulated as:

$$\sigma=(2q+1)(5N\log_2 N)+q(8N-n)+N \tag{10}$$

6- Using sliding window of size 1xn for the same matrix of 1xN pixels, q(2n-1)(N-n+1) computation steps are required when using CNNs for certain pattern detection or processing (n) input data. The theoretical speed up factor η can be evaluated as follows:

$$\eta = \frac{q(2n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \tag{11}$$

The theoretical speed up ratio, given by Eq. 11, with different sizes of input data and different in size weight matrices is listed in Table 4. Practical speed up ratio for manipulating matrices of different sizes and different in size weight matrices is listed in Table 5 using 1.7 GHz processor and MATLAB.

Table 4. The theoretical speed up ratio for detecting a certain pattern (n) in a stream of input data

Length of input matrix	Speed up ratio (n=400)	Speed up ratio (n=625)	Speed up ratio (n=900)
10000	5.1234	7.8224	10.9385
40000	4.6234	7.1865	10.2792
90000	4.3331	6.7566	9.7021
160000	4.1417	6.4653	9.2963
250000	4.0026	6.2513	8.9941
360000	3.8949	6.0847	8.7574
490000	3.8079	5.9497	8.5649
640000	3.7354	5.8371	8.4039
810000	3.6736	5.7409	8.2662
1000000	3.6199	5.6574	8.1464

Table 5. Practical speed up ratio for detecting a certain pattern (n) in a stream of input data

Length of input matrix	Speed up ratio (n=400)	Speed up ratio (n=625)	Speed up ratio (n=900)
10000	8.23	12.44	16.76
40000	8.01	11.98	15.63
90000	7.88	11.69	13.45
160000	7.54	11.42	12.87
250000	7.29	11.01	12.64
360000	7.02	10.73	12.37
490000	6.91	10.40	12.59
640000	6.83	10.17	12.13
810000	6.74	9.93	11.87
1000000	6.69	9.81	11.63

8 Conclusion

Fast parallel algorithms for finding the protein coding regions have been presented. A new approach for fast pattern detection has been presented. Such strategy has been realized by using our design for FNNs. Theoretical computations have shown that FNNs require fewer computation steps than conventional ones. This has been achieved by applying cross correlation in the frequency domain between the input data and the weights of neural networks. Simulation results have confirmed this proof by using MATLAB. The proposed algorithms can be applied successfully for solving many other bioinformatics problems like protein structure prediction, RNA structure prediction, promoter region identification, etc..

References

[1] Kiran Sree, P., Ramesh Babu, I.: Identification of Protein Coding Regions in Genomic DNA Using Unsupervised FMACA Based Pattern Classifier. International Journal of Computer Science & Network Security 8(1) (2008) ISSN: 1738-7906

[2] Kiran Sree, P., Ramachandran, R.: Identification of Protein Coding Regions in Genomic DNA Using Supervised Fuzzy Cellular Automata. International journal of Advances in Computer Science and Engineering 1 (2008) ISSN: 0973-6999

[3] Snyder, E.E., Stormo, G.D.: Identification of Protein Coding Regions In Genomic DNA. In: ICCS Transactions 2002 (2002)

[4] Audic, S., Claverie, J.-M.: Self-identification of protein-coding regions in microbial genomes, Structural and Genetic Information Laboratory, Centre National de la Recherche Scientifique-EP.91 (2002)

[5] Flocchini, P., Geurts, F., Mingarelli, A., Santoro, N.: Convergence and a periodicity in Fuzzy Cellular Automata: Revisiting Rule. Physica D 90 (2000)

[6] Maji, P., Chaudhuri, P.P.: FMACA: A Fuzzy Cellular Automata Based Pattern Classifier. In: Proceedings of 9th International Conference on Database Systems, Korea, pp. 494–505 (2004)

[7] Langton, C.G.: Self-reproduction in cellular automata. Physica D 10, 135–144 (2000)
[8] Toffoli, T.: Reversible computing. In: De Bakker, J.W., Van Leeuwen, J. (eds.) Automata, Languages and Programming, pp. 632–644 (1998)
[9] Vichniac, G.: Simulating physics with cellular automata. Physical D 10, 96–115 (1994)
[10] Chattopadhyay, S., Adhikari, S., Sengupta, S., Pal, M.: Highly regular, modular, and cascadable design of cellular automata-based pattern classifier. IEEE Trans. Very Large Scale Integr. Syst. 8(6) (2000)
[11] Fickett, J.: Recognition of protein coding regions in DNA sequences. Nucleic Acids Res. 10, 5303–5318 (1982)
[12] Farber, R., Lapedes, A., Sirotkin, K.: Determination of eukaryotic protein coding regions using neural networks and information theory. J. Mol. Biol. 226, 471–479 (1992)
[13] Bastin1, F., Malmedy, V.: A Retrospective Trust-Region Method for Unconstrained Optimization. Proc. Natl. Acad. Sci., USA 88, 11261–11265
[14] El-Bakry, H.M.: Face detection using fast neural networks and image decomposition. Neurocomputing Journal 48, 1039–1046 (2002)
[15] El-Bakry, H.M.: Human Iris Detection Using Fast Cooperative Modular Neural Nets and Image Decomposition. Machine Graphics & Vision Journal (MG&V) 11(4), 498–512 (2002)
[16] El-Bakry, H.M.: Automatic Human Face Recognition Using Modular Neural Networks. Machine Graphics & Vision Journal (MG&V) 10(1), 47–73 (2001)
[17] El-Bakry, H.M., Zhao, Q.: Fast Object/Face Detection Using Neural Networks and Fast Fourier Transform. International Journal of Signal Processing 1(3), 182–187 (2004)
[18] El-Bakry, H.M., Zhao, Q.: Fast Pattern Detection Using Normalized Neural Networks and Cross Correlation in the Frequency Domain. EURASIP Journal on Applied Signal Processing, Special Issue on Advances in Intelligent Vision Systems: Methods and Applications—Part I 2005(13), 2054–2060 (2060)
[19] El-Bakry, H.M., Zhao, Q.: A Fast Neural Algorithm for Serial Code Detection in a Stream of Sequential Data. International Journal of Information Technology 2(1), 71–90 (2005)
[20] El-Bakry, H.M.: New Faster Normalized Neural Networks for Sub-Matrix Detection using Cross Correlation in the Frequency Domain and Matrix Decomposition. Applied Soft Computing journal 8(2), 1131–1149 (2008)
[21] Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. Math. Comput. 19, 297–301 (1965)

Phase Transition in the Evolution of Artificial Life on Random Networks

Wei Qiang[1,2], Hui Li[1], and Hui Cao[3]

[1] School of Computer, China University of Geosciences
Wuhan 430074, People's Republic of China
[2] Faculty of Earch Resources, China University of Geosciences
Wuhan 430074, People's Republic of China
[3] Teaching and Experiment Center of Information Technology
China University of Geosciences
Wuhan 430074, People's Republic of China
qw@cug.edu.cn

Abstract. The cellular automation "game of life" on random networks is presented in order to count in topology effects on the evolution of artificial life. The average stationary density as functions of average degree and initial density are calculated, respectively. The present model exhibits a second order phase transition at an intermediate average degree $\langle k\rangle_C \simeq 8$ from a "sparse" state to a "dense" one. The average stationary density also reaches a small peak at the average degree $\langle k\rangle \simeq 4$. The average degree dependence of the phase transition point ρ_C is characterized by power laws.

Keywords: Cellular automata, Evolution, Artificial Life, Random Networks, Phase Transition.

1 Introduction

Cellular automata have been extensively studied because its relevant applications in many social, biological, and physical processes. It was originally introduced by von Neumann to study the logical properties of self-reproducing machine. Conway's game of life, introduced by Martin Gardener in 1970, is a two-dimensional cellular automaton, which was proved to be computation-universal and has been suggested to mimic aspects of complexity in nature [1,2].

The game of life was constructed on a universe of discrete space Z^2 (cells on a two-dimensional lattice) and discrete time t (an external clock) with symmetric rules governing local interaction. At each moment of time t, the sites on the lattices are occupied by a population of cells, $\{x_{ij}^t | i, j, t \in Z\}$. Each cell can be in one of two states 0 and 1, which are considered as "dead" or "live", respectively. The evolution is determined by the number of live cells among its eight nearest and next-nearest neighbors (orthogonal and diagonal): 1) each live site ($x_{ij}^t = 1$) will sustain the next time step if it has two or three live neighbors ($n_{ij}^t = 2, 3$), otherwise it will die by isolation (one or zero live neighbors) or overcrowding

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 501–509, 2008.
© Springer-Verlag Berlin Heidelberg 2008

(four or more live neighbors); 2) new life will be born at a dead site the next time step if there are exactly three live neighbors ($n_{ij}^t = 3$), otherwise it remains dead. The sum over the neighborhood, at each moment t, can be calculated as

$$n_{ij}^t = \sum_{r=-1}^{1} \sum_{s=-1}^{1} x_{i+r,j+s}^t - x_{ij}^t \ . \tag{1}$$

In a condensed form, the rule for the game of life is specified as follows:

$$n_{ij}^{t+1} = (x_{ij}^t \wedge (n_{ij}^t == 2)) \vee (n_{ij}^t == 3) \ , \tag{2}$$

where $==$, $\wedge$ and $\vee$ represent the logical operator EQUAL, AND, and OR, respectively [3]. The transitions of cells driven by unique clock take place synchronously. Starting from randomly initialized conditions, the game of life will evolve through complex patterns until reaching a stationary state with static or (time) periodic cell colonies. In spite of its simple algorithm, important characteristics are exhibited in the evolution of the game of life, including computational universality, construction universality, $1/f$ power spectrum and self-organized criticality [3,4].

Conway's game of life and relative work (deterministic or stochastic) construct cellular automata on two-dimensional lattice, which takes into account only local interactions with eight nearest and next-nearest neighbors. However, real populations rarely fall into this simple category because of the complex topology on which interactions exist [5,6]. The asymptotic density of live cells is also revealed transition with respect to the density of initial random configuration [7]. In this paper, we construct cellular automata on random network in order to investigate the effect of the topology on the evolution of life. The game of life is realized on population with random connections between individuals instead of population on regular lattice, that the interaction exceeds the geographical limit.

2 Model and Simulation

The interactions between the individuals of populations are described by a random network. The edge between two vertices represents that there exists interaction between two individuals which are located at these two vertices, respectively. To build random networks, we start with a set of N isolated vertices, each pair of which being connected with probability p. Self-connections and multiple connections are prohibited. Consequently the total number of edges n is a random variable with the expectation value

$$\langle n \rangle = \frac{N(N+1)}{2} p \ . \tag{3}$$

If G is a random network with N vertices and n edges, the probability of obtaining it by this network construction process is

$$\psi(G) = p^n (1-p)^{\frac{N(N-1)}{2} - n} \ . \tag{4}$$

By equation (3), the average degree of the random networks can be calculated as

$$\langle k \rangle = 2n/N = p(N-1) \simeq pN \ . \tag{5}$$

Random networks with various average degrees $\langle k \rangle$ can be obtained with this procedure.

The effect of topologies of populations on the game of life is investigated. Based on the built random network, we describe how the life evolves. Each vertex in the network may be in one of two states, representing the presence and absence of a live individual. The fate of each individual depends on its neighbors. In Conway's game of life, the number of neighbors is constant "eight" due to the underlying two-dimensional lattice. However, in the present model, we defined that two vertices are neighboring if there exists an edge between them. The number of neighbors can be varied over a wide range according to the construction of random networks. The rules of the evolution for the life are as follows: a dead individual will only come to life if it has exactly three live neighbors; a live individual will stay alive if it has two or three live neighbors. We start at time $t = 0$ with a random distribution of live individual with density ρ_0. At each time step, all vertices are updated synchronously according to the rule described above, until the game of life eventually settling down in a stationary state with static or (time) periodic cell colonies.

3 Results and Discussion

The random network, on which the game of life is defined, is characterized by the number of the vertices N and the probability p that there exists an edge between a pair of vertices. We have performed extensive numerical simulation with the described model on the random networks with different connection probabilities $p \in [0, 1]$ and sizes ranging from $N = 100$ to $N = 1000$. For every system with size N, the calculation results are averaged over both m different realizations of the network and 10 independent runs for each network realization, in such a way that $m \times N \simeq 2 \times 10^5$.

3.1 Random Network

Figure 1 shows the degree distributions that result from numerical simulation of random networks, which are generated with the number of vertices $N = 60000$ and the connection probability $p = 2.0 \times 10^{-3} \sim 4.4 \times 10^{-2}$. The number of vertices with degree k, N_k, is calculated and the ratio of N_k/Nversus k is plotted. The degree distribution $\psi(k)$ can be well approximated by Poisson distribution

$$\psi(k) \simeq e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k\,!} \ , \tag{6}$$

except for networks with very low connection probability, e.g. 2×10^{-3}, half of the distribution is truncated. It can be observed that with the increasing of connection probability p, the average distribution $\langle k \rangle$ increase, whereas the peaks of the distribution decrease and the degree of vertices have a wider distribution.

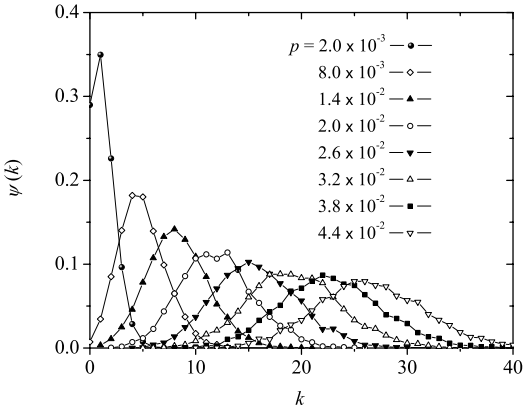


Fig. 1. Degree distribution of random networks, with $N = 6000$ and connection probability $p = 2.0 \times 10^{-3} \sim 4.4 \times 10^{-2}$

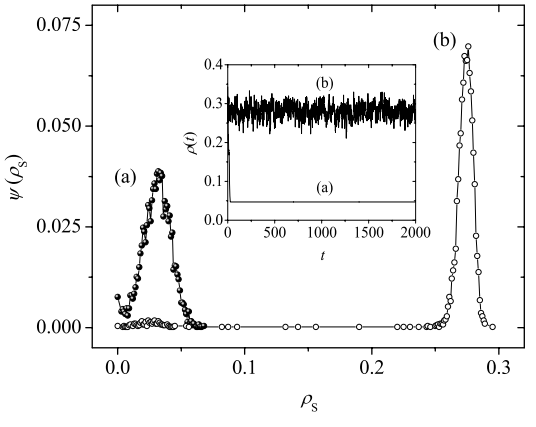


Fig. 2. The stationary density distribution of live individuals $\psi(\rho_S)$ with connection probability $p = 0.07$ (a) and $p = 0.08$ (b) for random network with $N = 1000$. Inset: the density of life $\rho(t)$ as a function of time t at two corresponding p values.

3.2 Game of Life

The effect of population topology, i.e. the network structure on the evolution of live individuals is investigated. The initial density is chosen to be $\rho_0 = 0.35$ without loss of generality. Figure 2 shows the stationary density distribution $\psi(\rho_S = \rho(t \to \infty))$ of live individuals with connection probability $p = 0.07$ and $p = 0.08$ for random network with $N = 1000$, respectively. It can be observed that the density of live individual $\rho(t)$ rapidly drops to a very small value at $p = 0.07$. However, the density $\rho(t)$ decreases a little and then approaches a large fluctuating value at $p = 0.08$. The stationary state of life has been reach within 1000 time steps. The stationary density distribution $\psi(\rho_S)$ at $p = 0.07$

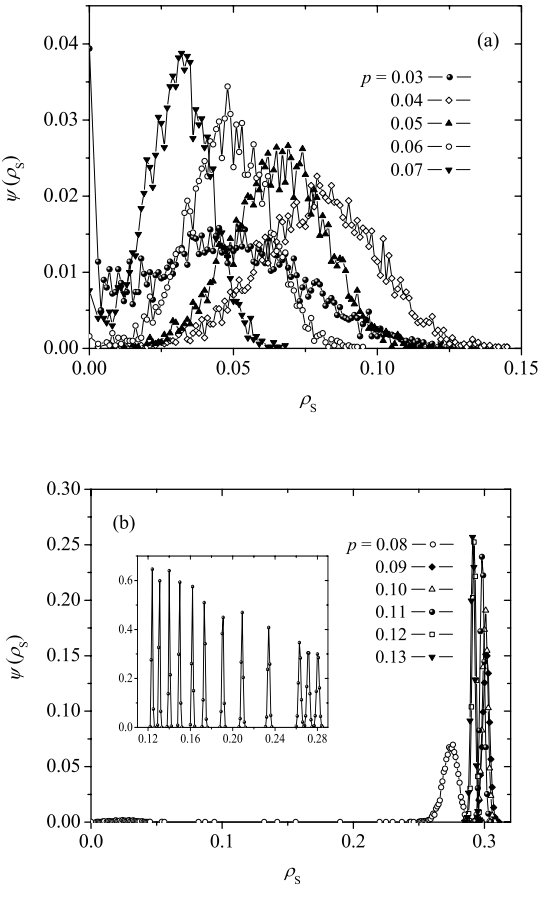


Fig. 3. The stationary density distribution of live individuals $\psi(\rho_S)$ for random network with $N = 1000$ and connection probability: (a) $p \leq 0.07$; (b) $p \geq 0.08$

and $p = 0.08$ are also plotted in figure 2, wherein peaks are easily identified. The positions of the peaks exhibit large difference corresponding to the two different p values. The significant change of density $\rho(t)$ and distribution $\psi(\rho_S)$ within such a small range of p indicated that there exits a phase transition of life.

In order to illustrate the network-induced phase transition of life, the stationary density distributions $\psi(\rho_S)$ of live individuals are calculated. Figure 3 presents the simulation results for random network with $N = 1000$ and connection probability from $p = 0.03$ to $p = 0.13$. It is observed that $\psi(\rho_S)$ has a normal-like distribution, except for $p \leq 0.03$, half of the distribution is truncated by axis of ordinates. The positions of the peaks, i.e. the average stationary density $\psi(\rho_S)$, fall into the range $\psi(\rho_S) < 0.06(5)$ for connection probability $p \leq 0.07$ (see Fig. 3(a)) and into the range $\psi(\rho_S) < 0.27(5)$ for connection probability $p \leq 0.08$ (see Fig. 3(b)).

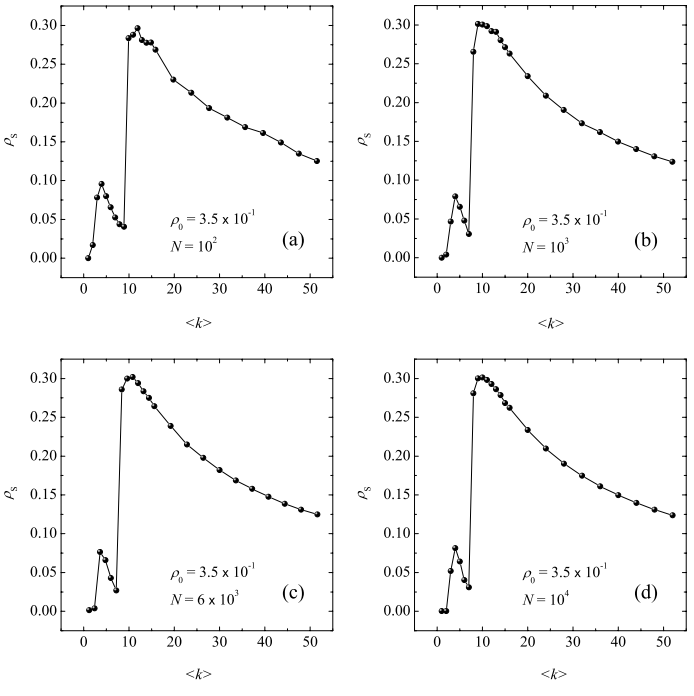


Fig. 4. The average stationary density of the life ρ_S as a function of average degree $\langle k \rangle$ for systems with initial density $\rho_0 = 0.35$ and different sizes: (a) $N = 10^2$; (b) $N = 10^3$; (c) $N = 6 \times 10^3$; (d) $N = 10^4$

The critical behavior of game of life is further investigated based on evidence provided for phase transition of stationary density at certain finite value $p \simeq 0.08$ aforementioned. Figure 4 shows the average stationary density of the life $\rho_S = \langle \rho(t \to \infty) \rangle$ as a function of average $\langle k \rangle$ degree for the system with sizes from $N = 10^2$ to $N = 10^4$. The initial density is chosen to be $\rho_0 = 0.35$ without loss of generality. The phase transition point of stationary density can be easily identified as average degree $\langle k \rangle_C = 8$ in spite of the system size N, wherein the number of neighbors is identical to those of Conway's game of life.

It is also interesting to find that ρ_S reaches a small peak at the average degree of random networks $\langle k \rangle \simeq 4$. The corresponding connection probability p_C can be determined by equation (5). In the definition of two-dimensional cellular automata, such as Conway's game of life, average degree of vertices $\langle k \rangle \simeq 4$ and $\langle k \rangle \simeq 8$ on regular lattice corresponds to Moore's neighbor and von Neumann's neighbor, respectively.

The dependence of asymptotic density $\rho_S = \langle \rho(t \to \infty) \rangle$ on the density $\rho_0$of a random uncorrelated configuration is investigated. Figure 5 presents simulation results with system size $N = 10^4$ and connection probabilities range from $p = 2 \times 10^{-4}$ to $p = 3.2 \times 10^{-3}$ (correspond to average degree from $\langle k \rangle = 4$ to $\langle k \rangle = 32$ by Eq. 5). It can be observed that ρ_S approaches zero in a ρ_0 range

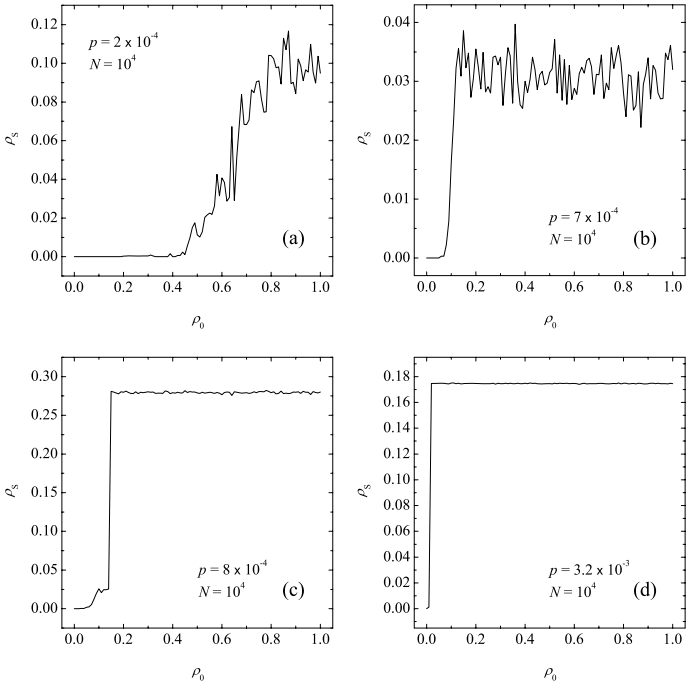


Fig. 5. The average stationary density of the life ρ_S as a function of initial density ρ_0 for systems with system size $N = 10^4$ and different connection probabilities: (a) $p = 2 \times 10^{-4}$; (b) $p = 7 \times 10^{-4}$; (c) $p = 8 \times 10^{-4}$; (d) $p = 3.2 \times 10^{-3}$

of $[0, \rho_C)$, and ρ_S begins to increase in the ρ_0 range of $[\rho_C, 1]$, wherein there are two categories that can be identified: $p < 8 \times 10^{-4}$ (see Fig. 5(a) and Fig. 5(b)) and $p \geq 8 \times 10^{-4}$ (see Fig. 5(c) and Fig. 5(d)). In the first category, ρ_S begins to increase with strong fluctuation from critical point ρ_C, and is correlated with first order phase transition. However, in the second category, ρ_S keeps roughly constant in a ρ_0 range of $[\rho_C, 1]$, and the phase transition is assumed to be a second order one.

In order to investigate the average degree dependence of the phase transition point ρ_C, $\rho_S \sim \rho_0$ relations for different connection probabilities are simulated. The phase transition point of initial density is determined by the position where ρ_S experience drastic change, increasing from zero vicinity to certain limited value. The critical initial density ρ_C as a function of the average degree $langlek\rangle$ for the system size $N = 1000$, together with the fitted lines that take double logarithmical forms are summarized in Figure 6. The order parameter is expected to have a power-law behavior

$$\rho_C \propto \langle k \rangle^{\beta}. \tag{7}$$

where β is the critical exponent of the phase transition. The power-law dependence for average degree $2 \leq \langle k \rangle \leq 8$ is summarized in Fig. 6(a) and $8 \leq \langle k \rangle \leq 52$

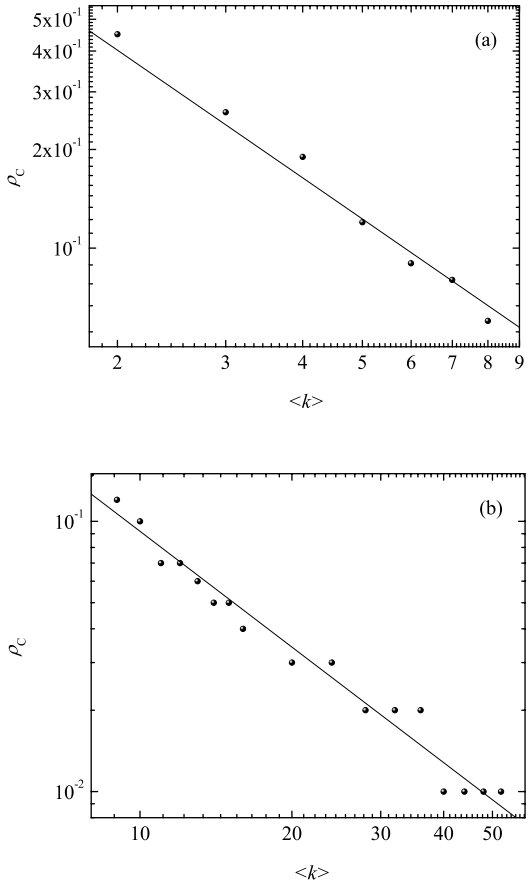


Fig. 6. The critical initial density ρ_C as a function of the average degree $\langle k \rangle$ for the system size $N = 1000$ on double logarithmic plots: (a) $2 \leq \langle k \rangle \leq 8$; (b) $8 \leq \langle k \rangle \leq 52$

in Fig. 6(b). The slope of the line fitted to the data can be associated with the critical exponent for which we obtained $\beta = -1.42 \pm 0.06$ with average degree $8 \leq \langle k \rangle \leq 52$, and $\beta = -1.30 \pm 0.06$ with average degree $2 \leq \langle k \rangle \leq 8$.

4 Conclusion

The cellular automation "game of life" on random networks has been investigated by extensive simulations. The results show that with the increase of the average degree $\langle k \rangle$ of random networks, the life exhibits a second order phase transition at an intermediate average degree $\langle k \rangle_C \simeq 8$. The average stationary density ρ_S, fall into the range $\rho_S < 0.06(5)$ for average degree $\langle k \rangle \leq 7$ and into the range $\rho_S > 0.27(5)$ for average degree $\langle k \rangle \geq 8$. The average stationary density ρ_S also reaches a small peak at the average degree $\langle k \rangle \simeq 4$. The critical average degree

corresponds to Moore's neighbor ($\langle k \rangle_C \simeq 8$) and von Neumann's neighbor ($\langle k \rangle \simeq 4$) of cellular automata on two-dimensional regular lattice. The average degree dependence of the phase transition point ρ_C is characterized by power laws, and the critical exponents are obtained. The present model is a generalization of Conway's game of life to random space, which paves a way to study the evolution of artificial life with complicated topology.

Acknowledgments. The work was supported in part by the China Postdoctoral Science Foundation under Grant No. 20080431011, and by the Research Foundation for Outstanding Young Teachers of China University of Geosciences under Grant No. CUGQNL0709.

References

1. Gardner, M.: The Fantastic Combinations of John Conway's New Solitaire Game "Life". Sci. Am. 223, 120–123 (1970)
2. Huang, S.Y., Zou, X.W., Tan, Z.J., Jin, Z.Z.: Network-Induced Nonequilibrium Phase Transition in the "Game of Life". Phys. Rev. E 67, 026107 (2003)
3. Andrecut, M.: 1/f Noise in the "Game of Life". Mod. Phys. Lett. 14, 53–57 (2000)
4. Alstrom, P.: Self-Organized Criticality in the "Game of Life". Phys. Rev. E 49, R2507–R2508 (1994)
5. Levene, M., Roussos, G.: A Two-Player Game of Life. Intl. J. Mod. Phys. C. 14, 195–201 (2003)
6. Blok, H.J., Bergersen, B.: Synchronous versus Asynchronous Updating in the "Game of Life". Phys. Rev. E. 59, 3876–3879 (1999)
7. Banoli, F., Rechtman, R., Ruffo, S.: Some Fact of Life. Physica A 171, 249–264 (1991)

A Novel Automatic Parameters Optimization Approach Based on Differential Evolution for Support Vector Regression

Jiewei Li and Zhihua Cai

Department of Computer Science, China University of Geosciences, Wuhan, China
Jieweili2006@gmail.com

Abstract. An appropriate parameters selection can significantly affect the accuracy of support vector regression (SVR) model. In this paper, a new evolutionary approach based on Differential Evolution (DE-SVR) is developed to train the SVR model. The approach evolves automatically the optimal model parameters by the differential mutation operations. Experimental results on several real-world datasets demonstrate that, comparing with the GA-based SVR and the Grid search methods, the DE-SVR can search the optimal parameters much more rapidly with less training time to build the SVR model, and has the comparable prediction accuracy as Grid search, even better than GA-based SVR. Therefore, the new evolutionary DE-SVM approach is an efficient method for automatic parameter determination of SVR problem.

Keywords: Support vector Regression, Differential Evolution algorithm, parameter optimization, predictive accuracy.

1 Introduction

Regression approximation of a given data set is a very common problem in a number of applications. Support vector machine (SVM) is a new method of machine learning based on VC-dimension. Originally, SVM has been developed to solve pattern recognition [1]. In recent years, SVM and kernel methods were successfully applied to solve high dimensional function estimation problems, developed by Vapnik (also called -SVR [2]). SVR adheres to the principle of structural risk minimization and can accurately fit the nonlinear regression datasets by kernel function.

Many extensions and improvement of SVM have been developed recently to enhance the performance for SVR. However, the hyper-parameters selection of SVR is still an unavailable problem. The quality of SVR models depends on the proper setting of hyper- parameters, including SVM kernel parameters, the regularization constant C and insensitive coefficient. To build an effective SVR model, these control parameters need to be set carefully ([3]Duan, Keerthi, 2001) to overcome the over-fitting or under-fitting. The main issue for practitioners trying to apply SVM regression is how to set these parameters to ensure good generalization performance, which no general guidelines are available so far.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 510–519, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The combination of evolutionary algorithms and SVM is appropriate, as training a SVR model is actually to solve a Quadratic Programming problem, which includes many uncertain parameters. Peng, Wu, and Goo [4] proposed a GA-SVR model for parameter optimization in SVR, and implemented it to forecast TWSI, wherever, not all the parameters was optimized the proposed method. Differential Evolution algorithm (DE) proposed by Storn and Price [5] is an efficient evolutionary optimization algorithm, and has been successfully suggested for constrained optimization by Lampinen [6], more algorithms within the DE framework have been put forward ([7], [8]), mainly because it has high convergence properties and has fewer control parameters than GA. It is extremely robust in locating the global minimum, which does not employ binary encoding like simple GA.

Therefore, considering the advantages of DE, in this study, we propose a new approach base on Differential Evolution (DE) algorithm to determine parameters of SVR, which can automatically optimizes all SVR's parameters simultaneously with the evolutionary process from the training data. The proposed approach was compared with the traditional Gird search approach and the GA-based SVR model.

2 Brief Introduction of Support Vector Regression (SVR)

Here, a brief introduction of SVR is given. For a data set $G = \{x_i, y_i\}$, $x_i \in R^n$, $y_i \in R$ denotes the input vectors and target values respectively. SVR model can be obtained by minimizing the sum of the empirical risk and model complexity.

SVM performs linear regression using a new type of loss function called ε-insensitive loss function proposed by Vapnik [1]. The constraint of ε-loss functions defines a tube with radius ε around the hypothetical regression function. If the data is positioned in this tube ε, the loss function can be regard as 0. By introducing the positive slack variables ξ_i and ξ_i^*, the SVM regression can be formulated to the following constrained optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \\ & y_i - w^T \phi(x_i) - b \le \varepsilon + \xi_i \\ \text{subject to} \quad & w^T \phi(x_i) + b - y_i \le \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \ge 0, i = 1, \cdots, l, \varepsilon > 0 \end{aligned} \tag{1}$$

Where $\phi(x)$ is the feature mapping in the high-dimensional feature space. w and b are the slope and offset of the regression line. C>0 called regularization parameter, is a positive constant. The positive slack variables ξ_i and ξ_i^* are to measure the deviation of training samples outside tube ε zone.

The constrained optimization problem given by Eq. (1) can be reformulated into dual problem formalism by introducing Lagrange multipliers [1] . Based on Karush-Kuhn-Tucker's (KKT) conditions, the function is given by:

$$f(x)=\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)k(x_i,x_j)+b. \tag{2}$$

Where $\alpha_i>0$, $\alpha_i^*>0$ are the Lagrange multipliers corresponding to the training data; $K(x_i,x_j)=\phi(x_i)\phi(x_j)$ denotes the kernel function, which satisfies the Mercer's conditions. The typical kernel functions contain the liner kernel, the polynomial kernel $(x_i^{\mathrm{T}}x_j+1)^d$, and the Gaussian kernel $\exp\left(-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2\right)$. The parameter σ represents the bandwidth of RBF kernel. This paper shows examples of SVR using radial basis function (RBF) kernel.

3 Differential Evolution

Differential Evolution (DE) is a new simple evolutionary optimization algorithm, which is an effective parallel direct search adaptive approach to global optimization. DE starts to explore the search space by randomly choosing the initial candidate solutions within the boundary, combines the crossover, mutation and selection to evolve the final individual solution. For each individual $x_{i,G}, i=1,\cdots,N$, where G denotes the current generation, DE's basic strategy can be described as follows:

- Mutation: DE randomly selects three individuals $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$ from the current set S. A mutant vector $\upsilon_{i,G+1}$ is generated using a *differential mutation* operation according to:

$$\upsilon_{i,G+1}=x_{r1,G}+F\cdot(x_{r2,G}-x_{r3,G}) \tag{3}$$

with random integer indexes $r_1,r_2,r_3\in\{1,\cdots,N\}$, and $r_1\neq r_2\neq r_3\neq i$;where F is a positive real number between [0, 1].

- Crossover: DE uses a binomial crossover operation, the mutated individual $\upsilon_{i,G+1}$ is mated with $x_{i,G}$ to generate the offspring or trial individual $u_{i,G+1}=(u_{1i,G+1},u_{2i,G+1},\cdots,u_{Di,G+1})$, using the crossover rules as Eq.(4).

$$u_{ji,G+1}=\begin{cases}\upsilon_{ji,G+1} & \text{if rand}(j)\le CR \text{ or } j=rnb(i)\\ x_{ji,G} & \text{if rand}(j)>CR \text{ or } j\neq rnb(i)\end{cases} \tag{4}$$

Where $\text{rand}(j)\in[0,1]$ is a uniform random number, CR is the crossover constant between [0, 1], and $rnb(i)$ is a randomly chosen index which ensures $u_{i,G+1}$ gets at least one parameter from $\upsilon_{i,G+1}$.

- Selection: The Selection operation employs a *knock-out* competition to decide the next generation, which can be expressed as follow：

$$u_{ji,G+1}=\begin{cases}\upsilon_{ji,G+1} & \text{if } f(u_{i,G+1})\le f(x_{i,G})\\ x_{i,G} & \text{otherwise}\end{cases} \tag{5}$$

The above steps of reproduction and selection are repeated generation after generation until the stopping criteria are satisfied.

4 Proposed DE-SVR Parameters Optimization

To construct an effective SVR model, a robust parameter selection or optimization strategy is a pre-requisite to obtain a well-performed SVM regression model. The generalization performance of the SVR model heavily depends on the appropriate setting of the control parameters, include the following: (1) regularization parameter C; (2) The kernel parameter: it reflects the distribution/range of x-values of training data. (3) The tube size of insensitive loss function: it can affect the number of support vectors used to construct the regression function. The three parameters, which are called hyper-parameter, is selected to be optimized in the SVR.

In this paper we present a DE-SVR parameters selection model for SVM regression. The new approach makes the best use of DE in optimizing all control parameters of SVR and searches dynamically for better combinations of these parameters by yielding a smaller mean absolute percentage error. The SVM model then performs the regression using these optimal values to improve the prediction efficiency. The process of DE-SVM optimizing the parameters was illustrated in Fig 3.

During the evolution, each new chromosome is evaluated by sending it to the SVM model. The optimal values of SVR are evolved by with a randomly generated initial populations consisting of chromosomes. The DE implements selection, crossover, and mutation operators to generate the offspring of the existing population. Selection operation employs a greedy competition to select excellent chromosomes to reproduce. Crossover and differential mutation operation were performed randomly to modify the chromosome to generate new individual. The solution process continues from one generation to another until termination criterion are satisfied.

Details design and model structure of our proposed DE-SVR approach is described as follows:

4.1 Chromosome Representation

Instead of binary coded representation, in this paper DE adopts the real-coded genomes with corresponding genetic operators performed. The values of parameters were coded to generate a chromosome. The chromosome X was represented as $X = \{p_1, p_2, p_3\}$ where *p1*, *p2* and *p3* denote the three parameters C, ε and σ^2 respectively. The bound values of the parameters are determined by users.

4.2 Fitness Function

The fitness of an individual of the population is the criterion for DE to optimize all the parameters. In this paper the k-fold cross validation ([3].Duan et al. 2001) method is used to evaluate fitness. The DE-SVR model is built using k-1 subsets which randomly divided by k-fold cross validation as the training set. The prediction accuracy on the last subset is described by root mean square error (*RMSE*) to measure the performance of the model in this generation, therefore, the fitness function is defined as the negative average *RMSE* of the k-fold cross validation (commonly k=5), which is denoted as follows:

$$fitness = -RMSE_{average} = -\frac{1}{5N}\sqrt{\sum_i (\hat{y}_i - y_i)^2} \times 100\% \quad (6)$$

Where $\hat{y}_i$ and y_i represent the actual and validation values respectively, N is the number of training samples. In the evolutionary process, the elitist strategy is employed to guarantee the best individual to be surviving. Therefore, the best individual with the larger $RMSE_{average}$ has a better chance to survive in the successive generations.

4.3 The Stopping Criteria

The DE evolution process was repeated until the stopping criterion is satisfied. In this paper the stopping criterion is set according to two criterions. While the difference of fitness value in every ten generations is less than or equal a given tiny value η, or the specified maximum number of generations is reached, the evolution process is terminated.

As shown in Figure 1, the procedure of the proposed DE-SVM approach can be briefly described as table1. DE uses the real-value code of parameters in populations without performing coding and encoding process before calculates the fitness values, which makes the evolutionary process more simple and rapidly.

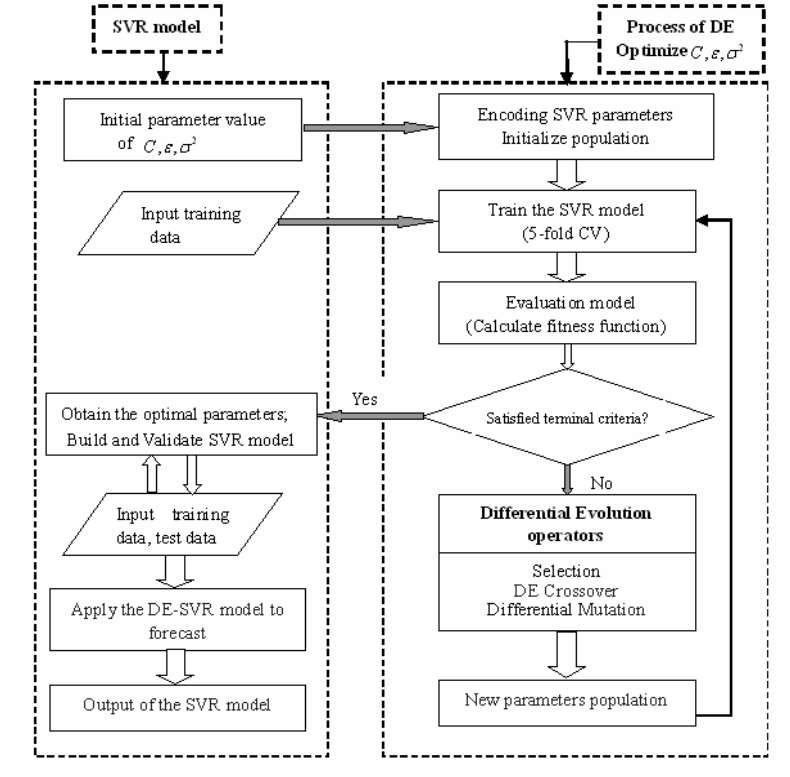


Fig. 1. The proposed DE-SVM parameters optimization model

Table 1. Steps of DE-SVM optimization model

Step1. Initialization: Set the initial parameter value of C,ε,σ^2 in SVR, encode them into chromosomes $X_i=\{C,\varepsilon,\sigma^2\}$; Initialize the correlative parameters of DE algorithm: population size (Popsize), the max generation, crossover probability (CR) and scale factor (F) of DE. Randomly initialize all the initial parameter population chromosomes of Popsize;
Step2. Using the training data and the initial parameter value to train the SVR model, compute the output $obj(X)$ of model;
Step3. Evaluate fitness: Calculate the current fitness value fitness (*j*) of the *j*th individual (*j*=1,…,Popsize) corresponding to 5-cross validation model in training set according to Eq.(6), and perform selection operation to find the best individual of current generation according to Eq.(5);
Step4. Calculate the difference $\lvert\Delta E\rvert$ of best fitness(*k*) and best fitness(*k+1*) between the two near generation; while the stopping criteria is not satisfied , go to next Step; otherwise, go to Step7;
Step5. Performing the Differential evolution, generate offspring population by performing crossover and mutation on parent pairs according to Eq.(4), (3);
Step6. Update the population; go back to step2;
Step7. Obtain the optimal parameters to train SVR model.

5 Experiment Result and Discussion

The research data used in this study to measure the performance of the proposed DE-SVR approach is obtained from the UCI benchmark machine learning repository. We tried several real world numeric datasets in the experiment which have been frequently used as benchmarks to compare the performance of different methods in many papers. Fourteen regression datasets were used. Table 2 describes the characteristics of these datasets.

Table 2. The description of Datasets from the UCI repository

Data sets	No. of attributes	No. of instances	Datasets	No. of attributes	No. of instances
auto93	23	93	housing	14	506
autoHorse	26	205	hungarian	14	294
autoMpg	8	398	meta	22	528
bodyfat	15	252	pharynx	12	195
breastTumor	10	286	quake	4	2178
cholesterol	14	303	sensory	12	576
cleveland	14	303	strike	7	625

The proposed DE-SVR approach was implemented on a PC with an Intel Pentium IV dual CPU 3.0GHz, 1024MB RAM, Windows XP operating system and the eclipse development environment by extending the Libsvm which is originally designed by Lin (2001)[13].

In this study, the parameter of DE was set according to R. Storn, K Price (1997) to obtain general autoregressive model. The 5-fold cross validation is employed. Scaling

was applied to prevent numerical difficulties in the calculation. The data values were scaled to the range [−1, +1]. Overview parameter values of this new approach were given in Table 3.

Table 3. An overview of some parameters' settings

DE parameter settings		SVR parameter settings	
Population size	20	C	[0.1-1000]
Number of generations	80	ε	[0.0001-3]
Selection	DE selection	σ	[0.1- 32]
Crossover probability CR	0.9	Stopping criterion η	0.001
Mutation factor F	0.5		

This research carries out the experiments using the proposed DE-SVR parameters optimization approach, and the results were compared with two existing methods: The GA-SVR approach and the Grid search. This study perform the GA-SVR approach to search the optimal using the simple GA.

In the experiments, the optimal parameter sets when the average *RMSE* of 5-fold cross validation of DE -SVR model is at its minimum. The searching process of optimal parameters was operated with 80 generations in total and the procedure run 10 times to obtain an average result of 10 times. Consequently, the *RMSE* of DE-SVR model shows a convergent state during the early generation evolution.

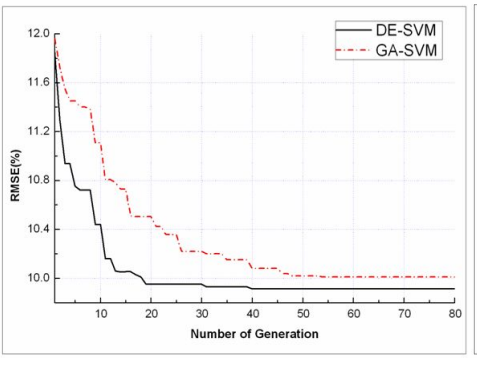


Fig. 2. The *RMSE* Optimization process of breast Tumor datasets in the evolutionary process

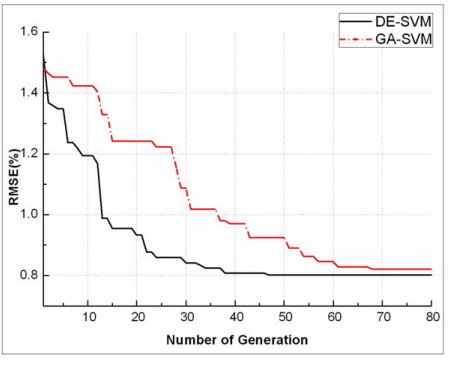


Fig. 3. The *RMSE* Optimization process of sensory datasets in the evolutionary process

Take two data set breastTumor、sensory for example, Fig. 2, Fig. 3 illustrates respectively the whole optimization process curves for the *RMSE* versus the generation number of the DE-SVR and GA-based SVR. In the evolutionary process, the *RMSE* in 5-cross validation in DE-SVR model decreased sharply with the increasing generation evolution. For the breastTumor dataset, when the evolution reached Generation 19, the average *RMSE* from the DE-SVR approach converged with the minimum *RMSE* value rated at 9.913%, while the SGA approach converges at Generation 45 with the minimum *RMSE* value rated at 10.137%. The other data sets have the similar optimization process curves for the DE-SVR and GA-based SVR approach. Therefore, the

average *RMSE* curves show that the convergence performance of DE-SVR approach outperforms obviously the GA-based SVR, and the proposed DE-SVR approach is featured with quite excellent efficiency.

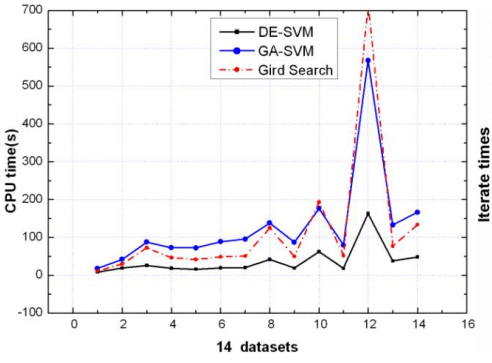


Fig. 4. The comparison of **CPU time** on the 14 datasets among DE-SVM, GA-SVM. and Gird search.

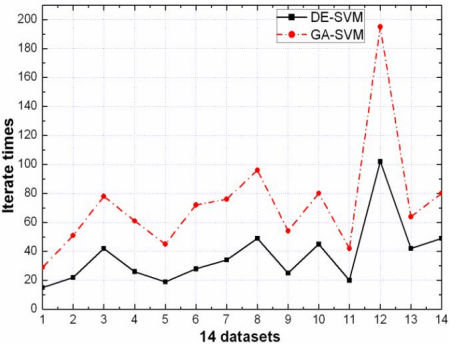


Fig. 5. The comparison of **Iterate times** on the 14 datasets between DE-SVM and GA-SVM

Table 4. Comparison results of DE-SVR, GA-based SVR and Gird search. Each approach runs ten times to calculate the average value. CPU time denotes the average training time. Iterate times denotes the average times. RMSE denote the minimum error among ten times.

Data sets	DE-SVR			GA-SVR			Gird Search		
	CPU time (s)	Iterate times	RMSE (%test)	CPU Time (s)	Iterate times	RMSE (%test)	CPU Time (s)	Iterate times	RMSE (%test)
auto93	8.4	15	9.156	17.7	23	9.168	11.4	42	9.147
autoHorse	18.9	22	39.296	42.1	41	39.204	29.6	66	38.761
autoMpg	25.7	42	7.623	87.5	78	7.879	72.3	131	7.641
bodyfat	18.4	26	8.327	72.6	61	8.196	46.7	115	8.257
breastTu-mor	15.7	19	9.913	72.4	45	10.137	41.6	102	10.013
cholesterol	19.4	28	49.462	88.6	72	51.227	48.4	119	48.953
cleveland	20.3	32	1.212	95.7	76	1.813	50.3	120	1.487
housing	41.6	49	8.775	138.2	96	9.243	125.2	148	9.073
hungarian	18.7	25	0.422	87.3	54	0.513	49.5	114	0.419
meta	62.3	45	489.045	176.3	80	482.202	192.8	172	475.415
pharynx	17.8	20	409.822	79.7	42	411.816	51.7	99	408.067
quake*	162.7	**102***	0.206	567.8	**195***	0.199	715.1	**427***	0.192
sensory	37.7	42	0.801	132.6	64	0.825	77.3	175	0.822
strike	48.1	49	496.235	166.3	80	501.087	133.9	195	504.734

Note: The parameter was set separately on the quake dataset because of its size. The evolutionary generation was set with 200 generations for running 5 times.

Fig.4,Fig.5 and Tables 4 show the corresponding comparison of our proposed DE-SVR approach, The GA-based SVR approach and the Grid search on testing dataset, respectively. The statistical metrics such as RMSE (root mean square error) was used as measures to evaluate the forecasting performance of the three models.

Based on the results obtained by Table 4, the DE-SVR approach cost the least time to obtain global the optimal SVR model on all of 14 data sets, while the average iterates times of GA-based SVR is nearly twice than that of our new approach on 9 data sets. The Grid search is the most time-consuming. With respect to the minimum error RMSE of model, the DE-SVR has the comparable prediction accuracy as Grid search, and has better accuracy than GA-based SVR on 10 datasets. Thus, the comparison results demonstrated that the overall performance of the proposed method outperforms other methods and had the best efficiency.

6 Conclusions

In this paper a novel automated parameter optimization procedure based on DE algorithm was presented for estimating the optimal SVR parameter for a training data set. The DE optimization algorithm, integrated using 5-fold cross validation, was first applied to obtain a set of optimal parameters which evolved the rapid and accurate SVR model parameter. Especially the 5-fold cross validation guaranteed high levels of performance and generalization ability during the optimization process of the SVR model.

The performance of the proposed approach had been evaluated by experiments on standard UCI regression datasets based on the RMSE measures. The results demonstrated that The DE-SVR optimization approach showed a quite faster convergence than GA-based SVR approach, and need much less time than the time-consuming Grid search approach. The results also indicated that, the new DE-SVR approach provided accurate results than the GA-based SVR model and performed much higher efficiency and improved performance. Hence, the proposed DE-base SVR approach, which can automatically tune all the parameters by searching rapidly the parameter space, and evolves a well-optimized SVR model with the lower prediction error, is a valid and promising alternative method for parameter optimization to solve SVM regression and forecasting problem.

References

1. Vapnik, V.N.: The nature of statistical learning theory. Springer, NewYork (1995)
2. Smola, A.J., et al.: A tutorial on support vector regression. Neuro COLT Technical Report Series, NC-TR-1998-030, Royal Holloway College, University of London, UK, vol. 10, pp. P 1–49 (1998)
3. Duan, K., Keerthi, S., Poo, A.: Evaluation of simple performance measures for tuning SVM hyperparameters (Technical report). Singapore: National Universityof Singapore, Department of Mechanical Engineering (2001)
4. Peng, K.L., Wu, C.H., Goo, Y.J.: The development of a new statistical technique for relating financial information to stock market returns. International Journal of Management 21(4), 492–505 (2004)

5. Storn, R., Price, K.: Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11(4), 341–359 (1997)
6. Lampinen, J.: A constraint handling approach for the differential evolution algorithm. In: Proc. 2002 IEEE Congress on Evolutionary Computation, Honolulu, Hawaii, May 2002, pp. 1468–1473 (2002)
7. Storn, R.: System design by constraint adaptation and differential evolution. IEEE Trans. Evol. Comput. 3(1), 22–34 (1999)
8. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evol. Comput. 4(1), 1–32 (1996)
9. Cherkassky, V., Mulier, F.: Learning from data: Concepts, theory, and methods. Wiley, New York (1998)
10. Scholkopf, B., Burges, J., Smola, A.: Advances in kernel methods: Support vector machine. MIT Press, Cambridge (1999)
11. Cristianini, N., Shawe-Taylor, J., Campell, C.: Dynamically adapting kernels in support vector machines. Advances in Neural Information Processing Systems 11, 204–210 (1998)
12. Deb, K., Goyal, M.: A combined genetic adaptive search (geneAS) for engineering design. Computer Science and Informatics 26(4), 30–45 (1996)
13. Murphy, P.M, Aha, D.W: UCI repository of machine learning datasets (1992), http://www.1ics.uci.edu/~mlearn/MLRepository
14. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2003), http://www.csie.ntu.edu.tw/~cjlin/libsvm
15. Bessaou, M., Siarry, P.: A genetic algorithm with real-value coding to optimize multimodal continuous functions. Struct. Multidisc. Optim. 23, 63–74 (2001)
16. Runarsson, T.P., Sigurdsson, S.: Model selection for support vector machines using an asynchronous parallel evolution strategy. In: IEEE International Conference on Neural Networks and Signal Processing, pp. 495–498 (2003)

An Improved Gene Expression Programming for Fuzzy Classification

Xiaobo Liu, Zhihua Cai, and Wenyin Gong

School of Computer science, China University of Geosciences, Wuhan 430074, China
Jerry-lxb@163.com

Abstract. Extracting accurate and understandable classification rules from data is a fundamental data mining activity. Fuzzy classification rules is considered a better classification of knowledge that the fuzzy rules of readability and analytical, and the use of fuzzy rules is very intuitive. In this paper, we present an improved gene expression programming (GEP) for extracting fuzzy classification rules by a logical operators instead of mathematical ones to represent the chromosome validity evaluation. Moreover, a novel technique to evaluate the fitness of the individual rather than transform the chromosome into expression tree is proposed. Our proposed approach has been tested on some benchmark datasets selected from UCI and the results indicate that our approach is highly comparable with other techniques including basic GEP, C4.5 and C4.5 rule.

Keywords: gene expression programming, fuzzy classification rules, logic operators, expression tree.

1 Introduction

Data mining [1] have many arrangements, the main arrangements includes: concepts/sorts depicts, associate analyze, classification and prediction, cluster analyze, and so on.

Classification system which based on rules has a disadvantage: for the continuous attributes, they have steep cut-off. Fuzzy classification system has the main feature is a clear collection in the classification system by adding fuzzy concept. Generally speaking, the classification of real events often contains a lot of uncertainty, fuzzy classification system is more suitable for the actual application of classification issues.

2 Gene Expression Programming

Gene expression programming (GEP) [2] is the most recent technique of evolutionary algorithm for data analysis. GEP uses fixed length, linear strings of chromosomes to represent computer programs in the form of expression trees of different shapes and sizes, and implements a GA to find the best program. The GEP is faster than GP [3].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 520–529, 2008.
© Springer-Verlag Berlin Heidelberg 2008

GEP has been used to extract classification rules in a mathematical form [3]. But the difficulty is that the classification rules generated by GEP in the mathematical form are usually barely comprehensible to the users and this is a serious problem for the user because it need great efforts to interpret this expression and to make it easy to be understood by the user. This is because the solutions may contain long expressions especially in the case of the problems that have a big number of attributes as we will see with microarray data.

The standard GEP was not developed to deal with different data types at one time. In fact, this limitation is due to the closure property [4], which has to be satisfied, that is, any non-terminal must be able to manipulate any data type returned from a terminal or nonterminal. In many applications, this is a serious limitation, leading to poor solutions to the problem dealt by GEP.

One method has been proposed in the paper to circumvent this limitation, allowing GEP to deal with different data types and satisfying the closure property without any problem.

The GEP algorithm proposed in this paper use atomic representation [5] to represent GEP gene in a logical form in the sense that it does not contains any syntactic constraints to be enforced by the system as in the case of enhanced version of genetic programming so that individuals can represent the rules that are valid and easy to interpret, due to the use of logical operators such as disjunctive, conjunctive and negation for the representation of the solution. This representation is easy to be applied in the case of nominal or binary attributes. However, it is not convenient to use continuous in this case. Therefore, we apply some mathematical mapping to transform these types of data to nominal or binary attributes.

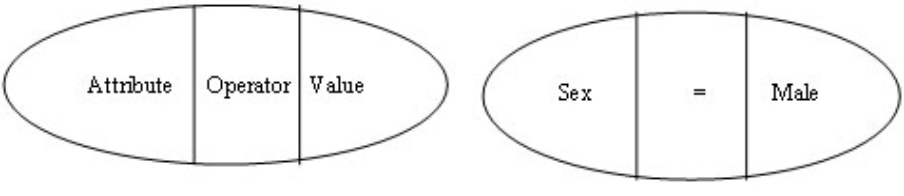


Fig. 1. Atomic Representation

Where each atom have three arguments, attribute name, relational operator (is),and attribute values of the data set being mined. It means that, the atom is syntactically a predicate of the form operator (var, operator, value) as in fig.1.

3 GEP for Discovering Fuzzy Classification Rules

3.1 Related Work

Cordon [6] proposed a genetic programming approach to extract classification rules, it is depending on a syntactic tree to represent the problem [7], see Fig.2. However, this representation is suffering from the closure problem, which has to be satisfied, that is, any non-terminal must be able to manipulate any data type returned from a terminal

or non-terminal. In many applications, this is a serious limitation, leading to poor solutions to the problem. Another two proposals can be found when using GP to evolve crisp classification rules set: (1) grammar-based on genetic programming and (2) Atomic representation.

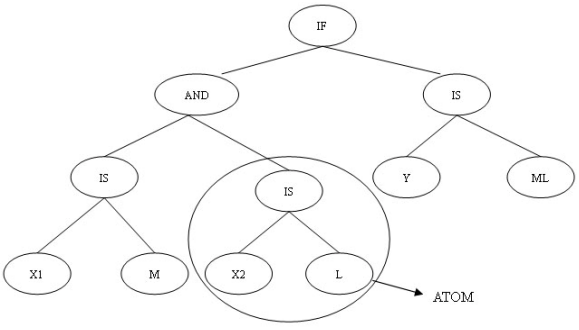


Fig. 2. A syntactic tree and the rule it represents If X1 is M and X2 is L then Y is ML

The question now arises, how do we interpret a statement like X is LOW and Y is HIGH or (not Z is MEDIUM). The standard definitions in fuzzy logic are:

$$\text{Truth (not x)} = 1.0 - \text{truth(x)} \tag{1}$$

$$\text{Truth(x and y)} = \text{minimum (truth(x), truth(y))} \tag{2}$$

$$\text{Truth(x or y)} = \text{maximum (truth(x), truth(y))} \tag{3}$$

Which are simple enough? Some researchers in fuzzy logic have explored the use of other interpretations of the AND and OR operations, but the definition for the NOT operation seems to be safe. Note that if you plug just the values zero and one into these definitions, you get the same truth tables as you would expect from conventional Boolean logic. The comprehensible of fuzzy rule-based system is related to various factors [8]

- Comprehensibility of fuzzy partition (i.e., linguistic interpretability of each fuzzy set, separation of neighboring fuzzy sets, the number of fuzzy sets for each variable)
- Simplicity of fuzzy rule (i.e., the number of input variables, the number of fuzzy if-then rules)
- Simplicity of fuzzy if-then rules (i.e., type of fuzzy if-then rules, the number of antecedent conditions in each fuzzy if-then rule).
- Simplicity of fuzzy reasoning (e.g., selection of a single winner rule, voting by multiple rules).

In this paper, we extract comprehensible fuzzy classification rules with gene expression programming. Atomic representation is used to solve the closure problem [6]. We extract one rule for each class, so the number of fuzzy rules that satisfy the condition number two is small. The length of the expression is small, so the antecedent expression is short, that satisfy condition three.

3.2 Data Fuzzification

In classification problem, training data may take either categorical (nominal) or numerical. In the case of category type, the fuzzification process is going straightforward. The values of the attributes are treated as fuzzy subsets. In this case, the fuzzy membership function is either zero or one.

In the case of numerical attribute, the data will be fuzzified into linguistic terms. In fact, the fuzzification in this case is a process of conceptualization, which is often used by people to reduce information overload in decision making process. For example, the numerical salary data may be perceived in linguistic terms such as high, average, and low. Their membership functions can be approximated based on expert's opinion or people's common perception. The membership may be derived from statistical data or from fuzzy clustering based on self-organized-learning or by GA [9].

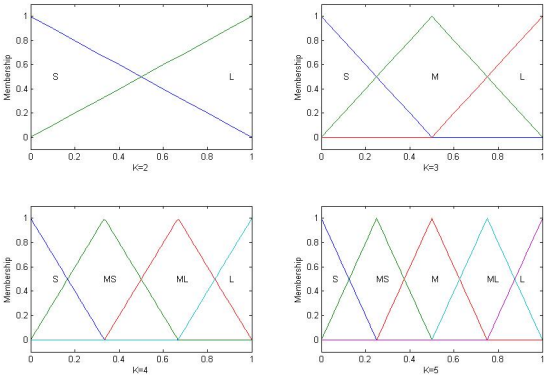


Fig. 3. Triangular membership function

In our work here, we use a simple method to fuzzyify the numerical attributes [9]. Let us assume that each numerical attribute (A) is partitioned into K fuzzy subsets{ A_1^k, A_2^k, …, A_k^k }, where A_i^k is a fuzzy linguistic variable (the subscript K is attached to indicate the number of fuzzy subset of each attribute). We can use any type of membership function (e.g. triangular, trapezoid, and exponential) form. In our model, the symmetric triangular membership function is employed for A_i^k , i=1, 2,…, K, as shown in Fig.3. Where each numerical attribute is normalized into interval [0, 1] by

$$A'= (A-\min (A))/ (\max (A)-\min (A)), \tag{4}$$

where A' is a new value of A, max(A) and min(A) are maximum and minimum value of attribute A respectively. The value of membership function μ_i^k for each linguistic variable A_i^k

$$\mu_i^k = \max\{1-(|x-a_i^k|/b_i^k), 0\} \tag{5}$$

where

$$a_i^k = (i-1)/(K-1), i=1,2,\ldots, K, \tag{6}$$

$$b_i^k = 1/(K-1). \tag{7}$$

3.3 Fuzzy Classification Rules

In our approach, the antecedent condition is expressed as fuzzy logic expression, and extract rule for each class using GEP. Since GEP can extract one rule in each run, to extract n-rule; therefore, we running GEP n- times (Mitchign approach).

The formula of the expression takes the form of fuzzy logic expression. The closure problem is one of the most challenge problems in extract classification rules especially in the case of nominal attributes (fuzzy linguistic value). Atomic representation is our key idea to treat the closure problem. The atom operator (var, operator, value) returns the value of membership function of fuzzy attribute fvar. Fig.4 shows an example of atomic representation, let chromosome length=9, four points in the head and five points in the tail, the rule can be expressed as

(X1 is high) and ((X2 is mid) or (X3 is low)).

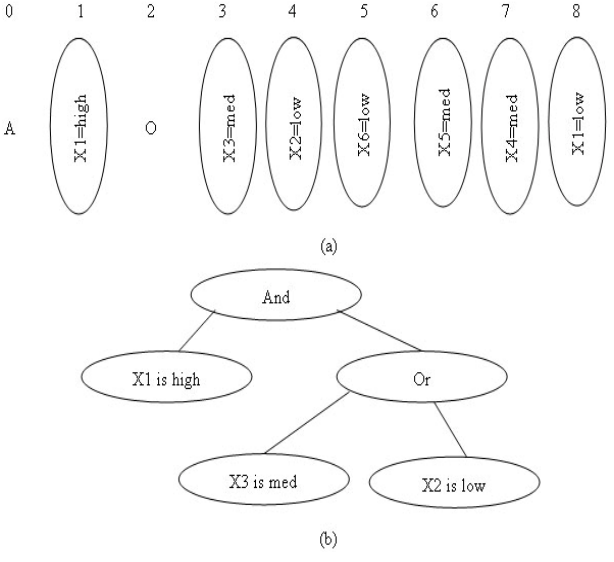


Fig. 4. The proposed model extracting fuzzy classification rules

Suppose the example has 0.75, 0.34, 0.2 for X1, X2, X3 respectively. Suppose also that the membership function for linguistic term high, med, low is 0.8, 0.7, 0.6 respectively. The expression becomes min (0.8; max (0.7; 0.6)), which return 0.7. This new representation lead to new operator called atomic mutation [8]. The atomic mutation change attribute by new attribute name or attribute value by new value as in algorithm 1.

```
Algorithm 1: Atomic mutation
Begin
  If (Random Number() > 0.5) then
     Replace fuzzy attribute with new attribute, and select
     random linguistic variable of new attribute
  Else
     Change fuzzy linguistic value to new another linguistic
     value
  End
```

3.4 Fitness Function

The main goal of defining fitness function is to measure the quality of the rules. To evaluate the fitness function, let A is antecedent of given rule. Once the degree to which each example belongs A has been computed, the predictive accuracy of i^{th} individual (fuzzy rule), denote Acc(i), is computed by formula [10].

$$Acc(i) = \frac{Corr\,\mathrm{Pr}ed - 1/2}{Tot\,\mathrm{Pr}ed} \tag{8}$$

In this formula, CorrPred(number of correct predictions) is the summation of degree of membership in A for all examples that belongs class positive. TotPred (total number of prediction) is the summation degree for all examples. The rational for subtraction from CorrPred in numerator is to penalize rules that are too specific, which are probably overfited to the data. For instance suppose CorrPred = 1, and TotP red = 1. without subtraction from CorrPred, the modified formula would return a predictive accuracy of 100% for the rule, which intuitive is over-optimistic estimate of predictive accuracy in this case. However, subtraction 1/2 from CorrPred the above formula returns 50%, which seems a more possible estimated of predictive accuracy. Clearly the value of CorrPred and TotP red the subtraction of 1/2 will not have a significant influence in value returned by formula, so this subtraction penalizes only rules that very specific, covering just a few examples.

4 A New Improved GEP

4.1 BGEP Compute the Individual's Fitness Value

Think the gene which composed by {A, O, N, X1=high, X2=low, X3=med, X4=med, X5=high, X6=low}, where 'A' mean AND, 'O' mean OR,'N' mean NOT. Evidently, in this example, n=2, if we propose h=4, then t=5, and the length of gene is 4+5=9.

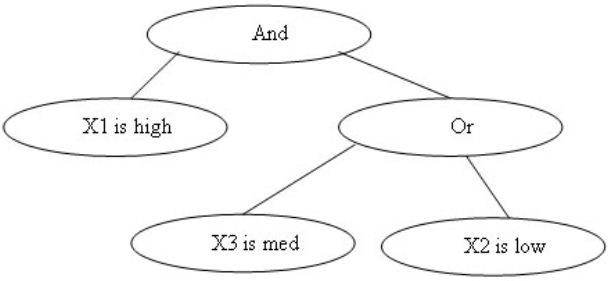


Fig. 5. Expression Tree

For example, think a gene:

A(X1=high)O(X3=med)(X2=low)(X6=low)(X5=high)(X4=med)(X1=low) (a)

Transfer this gene into an Expression Tree (ET), just order by from left to right, from top to down read the symbols of gene. Use this method the gene can transfer to expression tree as fig.5.

4.2 Improved GEP Compute Individual's Fitness Value

In this paper, we have designed a new way to get the fitness value, without traversing the tree or heap stack. But with this new method we must know another conception of effective gene. The genes of GEP consists of a head and a tail, but when initial population by roulette, the gene symbols of head and tail are not all useful, this is just for ensuring that every chromosome can properly express a right expression tree. For example the gene get from section 1, it's effective length is 11. So CHEN An-sheng, *et al* proposed a new way to compute the effective length of gene [11], and JANG Da-zhi, *et al* proposed a (Gene Read & compute machine) GRCM method to compute the fitness value of gene [12]. According to their method, we proposed a new algorithm without transforming the chromosome into expression trees and computing the effective length of gene, just building two arrays stack the symbol.

As the gene mention in section 1, we operator as follow:

A	X1=high	O	X3=med	X2=low	X6=low	X5=high	X4=med	X1=low

From the first place of gene, count=1, the first symbol is A, then count=2, put into array a:

A				

The second symbol is X1=high, then count=1, put into array b:

X1=high				

The third symbol is O, then count=2, put into array a:

A	O			

The forth symbol is X3=med, then count=1, put into array b:

X1=high	X3=med			

The fifth symbol is X2=low, then count=0, put into array b:

X1=high	X3=med	X2=low		

So the array a and the array b is:

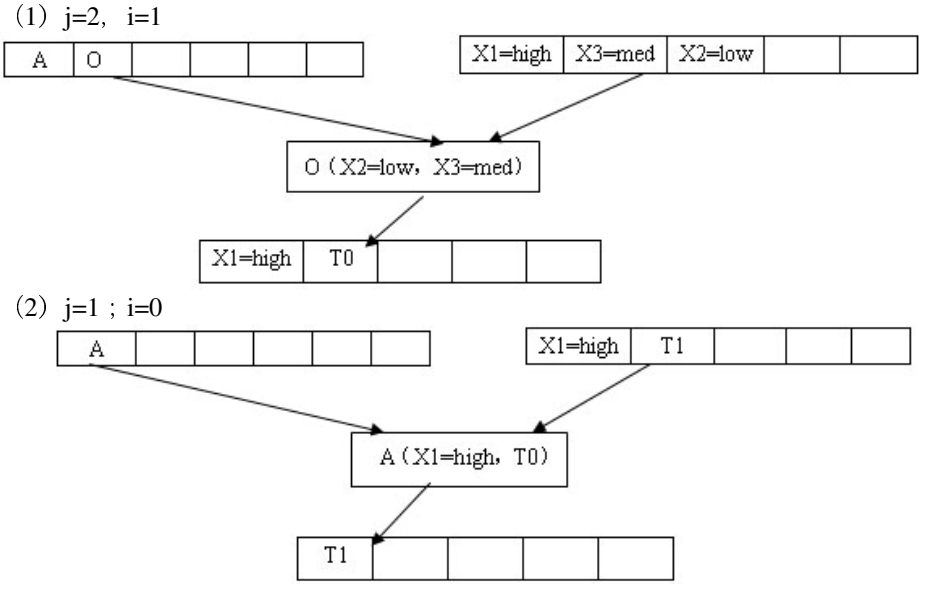
a[]:

A	O			

b[]:

X1=high	X3=med	X2=low		

Computer the fitness value:



Where T1 is the return value of gene expression programming, and T0 is value of (X2 is mid) or (X3 is low).

It is worth noticing that this method doesn't need to transfer the gene into tree and need not compute the effective length of gene, and this method have the better result than other methods on the fitness value, and easier than others. No matter which way we use to improve GEP, we must keep the legality of gene of GEP, that is, every gene of GEP have to use only one tree to express, so we can use this way to get the fitness value of chromosomes.

5 Experimental Results

In order to evaluate our technique, we have applied it to some standard datasets selected from UCI [13]. The criterion of chose data are: (1) GEP can get high classify

precision on those data; (2) because the based GEP classifier function fitness can not deal with the data style severe unbalance question, so we did not think about the severe unbalance question(eg. Balance Scale). To evaluate the performance of the classifier, accuracy is computed by 5-fold cross-validation method by dividing database into five partition where each partition contains a proximately proportion of labels as in the original dataset. In each iteration of the cross-validation, one of the five partitions was used as test and the other four partitions were used as training dataset. The predictive accuracy represents as average over all five iterations of cross-validation procedure. In order to extract a comprehensible rules, the rule must be simple. GEP provides the control of the size of chromosome by setting the head parameter a small number. Therefore, we can use GEP for extract comprehensible rule as we will explain in our computer results.

Table 1. Parameter settings

Population size	100	2-Point recombination rate	0.5
Number of fitness cases	16	Gene point recombination rate	0.1
Function set	A,O,N	IS transposition rate	0.1
Gene length	13	IS elements length	1,2,3
Number of gene	3	RIS transposition rate	0.1
Linking function	A	RIS elements length	1,2,3
Chromosome length	39	Gene transposition rate	0.1
Mutation rate	0.044	Selection range	100
1-Point recombination rate	0.2	Precision	0.01

Table 2. Comparison of BGEP, C4.5, C4.5 rule and our approach

Data Name	Example Number	Attribute Number	Accuracy average value (%)			
			BGEP	C4.5	C4.5Rule	Our Approach
Breast Cancer W.	625	9	95.9	94.6	95.5	96.1
Ionosphere	351	34	87.1	88.4	88.6	88.5
Iris-Setosa	150	4	99.7	99.3	95.8	97.9
IrisVersicolor	150	4	95.6	94.9	96.3	98.2
Iris-Virginica	150	4	94.7	94.2	96.1	98.1
Wine-Class1	178	13	94.7	94.2	93.7	94.5
Wine-Class2	178	13	92.7	92.4	92.8	93.2
Wine-Class3	178	13	90.8	88.4	90.0	91.2

6 Conclusion

We have proposed an improved GEP for discovering interesting fuzzy classification rules. Adopted atomic representation settle the question of GEP deal with different type of data and the closure problem, use array storage data avoided building Expression Tree, neglect compute the effective length of gene, reduced complexity and saved much time. From the mean feature of the extracted rules we can note that, the rules discovered by the improved GEP much more accurate than the rules discovered by basic GEP, C4.5 and C4.5 rule.

Acknowledgments

This work was supported by the Humanities Base Project of Hubei Province (Grant No. 2004B0011) and the Natural Science Foundation of Hubei Province (Grant No. 2003ABA043).

References

[1] Mitchell, T.: Machine Learning. McGraw-Hill Press, NewYork (1997)
[2] Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems 13(2), 87–129 (2001)
[3] Zhou, C., Xiao, W., Tirpak, T.M., Nelson, P.C.: Evolving accurate and compact classification rules with gene expression programming. IEEE Transactions on Evolutionary Computation (in review, 2002)
[4] Koka, J.: Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge (1992)
[5] Marghny, H., El-Semman, I.E.: Extracting fuzzy classification rules with gene expression programming. In: ALML 2005 Conference (2005)
[6] Cordon, O., Gomide, F., Herrere, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Sets and Systems 141(l), 5–31 (2004)
[7] Bojarczuk, C.C., Lopes, H.S., Freitas, A., et al.: A constrained-syntax genetic programming system for discovering classification rules: applications to medical datasets. Artificial Intelligence in Medicine 30(1), 27–48 (2004)
[8] Eggermont, J., Eiben, A.E., van Hemert, J.I.: A comparison of genetic programming variants for data classification. In: Hand, D.J., Kok, J.N., Berthold, M.R. (eds.) IDA 1999. LNCS, vol. 1642, pp. 281–291. Springer, Heidelberg (1999)
[9] Ishibuchi, H., Yamamoto, Y.: Fuzzy rule selection by multi-objective genetic local search algorithms and rule evolution measure in data mining. Fuzzy set and systems 141, 58–88 (2004)
[10] Romao, W., Freitas, A.A., Gimenes, I.S.: Discovery interesting knowledge from science and technology database with a genetic algorithm. Applied soft computing 4(3), 121–137 (2004)
[11] Jiang, D.-z., Wu, Z.-j.: New Method Used in Gene Expression Programming:GRCM 18(6), 1466–1468 (June 2006) (in Chinese)
[12] Chen, A.-s., Cai, Z.-h.: A New Decoding Method of GEP AND Its Application [J]. In: China National Computer Conference 2005, Tsinghua University Press (2005) (in Chinese)
[13] UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/`

An Improved K-Means Clustering Algorithm Based on Spectral Method

Shengwen Tian, Hongyong Yang, Yilei Wang, and Ali Li

College of Compute Science and Technology, LuDong University,
264025, Yantai, China
gltsw@yahoo.com.cn, hyyang@yeah.net,
wang_yilei2000@163.com, lliiaallii_2000@sohu.com

Abstract. It is well known that K-means algorithm is very sensitive to outliers, and often terminates at a local optimum. Furthermore, it is necessary for K-means algorithm to determine the number *K* of clusters as a priori knowledge in advance. Therefore, the quality of the result is not satisfactory. In this paper, we develop an improved K-means clustering algorithm—NK-means. NK-means is based on spectral methods, namely uses Normal matrix that is used in spectral analysis approaches to normalize original datasets, and then finds clusters in the processed datasets by K-means algorithm. We also propose a measure for the strength of clusters structure found by NK-means algorithm, which gives us an objective metric for choosing the number *K* of clusters into which a data set should be divided. Experiment shows that NK-means algorithm significantly outperforms K-means in the efficiency and accuracy.

Keywords: Clustering, K-means algorithm, Spectral Method, Measure.

1 Introduction

Clustering[1][2] is widely studied in data mining and considered the most important unsupervised learning problem. It is used to partition data set into clusters so that intra-cluster data are similar and inter-cluster data are dissimilar. Clustering technique is widely used in applications of financial data classification, spatial data processing, satellite photo analysis, and medical figure auto-detection etc.. Several major data clustering approaches are presented, including partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. K-means[3] is one of the simplest partition-based clustering algorithms that solve the well known clustering problem, and is linear to the size of the data set. However, K-means algorithm is usually sensitive to outliers, and often terminates at a local optimum, and it is necessary for K-means algorithm to determine the number *k* of clusters as a priori knowledge in advance. Therefore, the quality of the result is not satisfactory. Furthermore, it is usually designed as a memory-resident algorithm, which limits the scalability. According to the disadvantages of K-means algorithm, some optimization methods are developed[2], but most of them focus on the selection of *K* centroids. In this paper, we develop a novel method to optimize K-means algorithm with

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 530–536, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Normal matrix used in spectral analysis approaches, and obtain an improved K-means algorithm—NK-means, and also propose a measure for the strength of clusters structure, which gives us an objective metric for choosing the number *K* of clusters into which a data set should be divided.

This paper is organized as follows. After a brief review the structure and the properties of Normal matrix in section 2, the proposed new algorithm NK-means is described in section 3. And in section 4, we consider ways of determining when a particular division of a data set into clusters is a good one, allowing us to quantify the success of our clustering algorithms. We evaluate NK-means algorithm using a real data set whose expected cluster structures are known to us. The experiment results are presented in section 5. We conclude the paper with some future work in section 6.

2 Structure and Properties of Normal Matrix

Let *G* is an undirected graph with *n* nodes, let *A* denote the adjacency matrix of the graph *G*, whose element a_{ij} is equal to 1 or 0 if *G* is an unweighted graph and the weight w_{ij} of the edge between node *i* and node *j* otherwise. *G*'s Normal matrix[4] *N* is defined as $N = D^{-1}A$, where *D* is the diagonal matrix with elements $d_{ii} = \sum_{j=1}^{n} a_{ij}$, so, *N* is a symmetric matrix. The matrix *N* has always the largest eigenvalue equal to one, associated to a trivial constant eigenvector, due to row normalization. In a data set with an apparent cluster structure, *N* has also a certain number *k*-1 of eigenvalues close to one, where is the number of well defined clusters, the remaining eigenvalues lying a gap away from one. The eigenvectors associated to these first *k*-1 nontrivial eigenvalues, also have a characteristic structure: the components corresponding to nodes within the same cluster have very similar values x_i, so that, as long as the partition is sufficiently sharp, the profile of each eigenvector, sorted by components, is step-like. The number of steps in the profile corresponds again the number *K* of clusters.

To resolve the eigenvectors of Normal matrix *N*, Capocci[5] recast the eigenproblem into an optimization problem. For the weighted matrix *W*, whose elements w_{ij} are assigned the intensity of the link(*i*, *j*). The optimization method is as follows:

$$\text{Objective function:} \quad z(x) = \frac{1}{2}\sum_{i,j=1}^{n}(x_i - x_j)^2 w_{ij}. \tag{1}$$

$$\text{Constraint condition:} \quad \sum_{i,j=1}^{n} x_i x_j m_{ij} = 1. \tag{2}$$

where *n* is the number of the nodes in network, x_i are values assigned to the nodes, m_{ij} are elements of a given symmetric matrix *M*. the stationary points of *z* over all *x* subject to the constraint condition (2) are the solutions of $(D-W)x = \mu Mx$, where *D* is the diagonal matrix, $D = (d_{ij})$, $d_{ij} = \delta_{ij}\sum_{k=1}^{n} w_{ik}$, and μ is a Lagrange multiplier.

Different choices of the constraint M leads to different eigenvalues problem: for example choosing $M = D$ leads the eigenvalues problem $D^{-1}Wx = (1-2\mu)x$, and corresponds to the eigenproblem for the Normal matrix $N = D^{-1}W$.

However, the study of the eigenvectors profiles and the eigenvalues has practical use only when a clear partition exists, which is rarely the case. In the most common occurrences, the number of nodes is too large and the separation between the different clusters is rather smooth. Thus it is difficult to determine the number K of clusters, but it is easy to find an eigenvalue E ($E \neq 1$) that is closer to one than the others. The eigenvectors of E has the characteristic structure mentioned above. NK-means algorithm proposed in this paper makes use of the characteristic structure of the eigenvectors of E to transform original datasets by Normal matrix.

3 NK-Means Algorithm

Let a data set $T=\{x_1, x_2, x_3, \cdots, x_n\}$, in order to describe NK-means algorithm easily, we call data objects as nodes. Node x_i can be expressed as follows: $x_i=\{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{im}\}$, where x_{ip} $(1 \le i \le n, 1 \le p \le m)$ is an attribute value of node x_i. In NK-means algorithm, we first calculate the similarity S_{ij} between node x_i and node x_j by euclidean distance (3) or angle-off cosine methods (4) selected base on the concrete condition, and set adjacency matrix $W=(S_{ij})$ and diagonal matrix $D = (d_{ii})$, $d_{ii} = \sum_{j=1}^{n} a_{ij}$ to construct Normal matrix $N=D^{-1}W$. Then, the first non-trivial eigenvalue E which is the closest to one but not equal to one and the eigenvectors of the eigenvalue E are solved by Normal matrix N. Lastly, the eigenvectors are considered as the new data set and partitioned into K clusters using K-means algorithm.

$$S_{ij} = 1 \Big/ \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{im} - x_{jm})^2}\,. \tag{3}$$

$$S_{ij} = \frac{\sum_{k=1}^{m} (x_{ik} \times x_{jk})}{\sqrt{\sum_{k=1}^{m} x_{ik}^2} \times \sqrt{\sum_{k=1}^{m} x_{jk}^2}}\,. \tag{4}$$

The pseudo-code of NK-means algorithm is presented as follows:

Input: data set T and the number of clusters K;

Output: clustering $\{C_1, C_2, \ldots, C_k\}$;

Step1: calculate the similarity S_{ij} between node i and node j by original data set T;

Step2: get the adjacency matrix W and the diagonal matrix D by S_{ij}, and to construct Normal matrix $N = D^{-1}W$;

Step3: get the eigenvectors V associated to the first non-trivial eigenvalue E that is the closest to one but not equal to one in $N = D^{-1}W$;

Step4: select K nodes as initial cluster centroids randomly from V;

Step5: for each object v_i in V, assign it to the closest cluster C_j;

Step6: for each cluster C_i, update its center by averaging all of the objects v_j that have been assigned to it;

Step7: repeat S*tep5* and S*tep6* until convergence.

The traditional K-means algorithm partitions clusters by the distance similarity. It inclines to recognize the spheral clusters which are similar in size. When normalizing the data by Normal matrix *N*, the data is transformed from a matrix into a vector, the structures of the clusters are changed from three-dimensional sphere into two-dimensional circle. So, NK-means algorithm has the advantage as follows: 1).the influences of outliers can be reduced in the processed data; 2).the disadvantage of local optimum and the limitation of the scalability can be improved significantly.

4 Quantifying the Number K of Clusters

It is well known that the number of clusters is not known in practical situations ahead of time. This raises a new problem: how to we know which of these partitions are the best one for a given data set? Whether or not the clusters found by the algorithm is meaningful? To answer these questions we now define a measure of the quality of a particular division of a data set, which we call the *modularity*. Consider a particular division of a data set *T* into *R* clusters. Let *n* is the total number of the objects in *T*, x_i ($1 \le i \le n$) is the i^{th} object in *T*, C_{k} ($1 \le \text{k} \le R$) is the k^{th} cluster, $\overline{X}$ is a mean vector, $\overline{X}_k$ is the mean vector of C_{k}, $\|x\|$ denotes the euclidean distance. So, the sum of squares of deviations of *T* is defined as $F = \sum_{i=1}^{n} \|x_i - \overline{X}\|^2$, the sum of squares of deviations of C_{k} is $U_k = \sum_{i \in C_k} \|x_i - \overline{X}_k\|^2$, $P_R = \sum_{k=1}^{R} U_k$ represents the total sum of squares of deviations in *R* clusters.

It is clear that a good division into *R* clusters should have a low value of P_R, that is, the more reasonable the number of clusters is, the higher the value of F-P_R will be. The value of F-P_R on its own, however, is not a good indicator of the quality of the division since, for example, partitioning each object in *T* as a single cluster (i.e., $R=n$) would give the minimal value of $P_R=0$, while giving no information about clusters structure at all. Thus we can define a modularity measure by

$$Q = \frac{(F - P_R)/R}{F}. \tag{5}$$

Considering the influence of *R* on *Q*, Eq. (5), we define F-P_R is divided by *R* in *Q* function. The denominator *F* in *Q* function is used to normalize the value of *Q*, which makes the intervals of the value of *Q* are 0～1.

When we partition a data set *T* by NK-means algorithm or K-means algorithm, to set the number of clusters $K=2,3,\cdots,\text{n}-1$ and partition the objects in *T* into *K* clusters in turn according to the value *K* firstly, then we will calculate *Q* for each split of a data set into clusters according to different the value of *K* and look for local peaks in its value, which indicate particularly satisfactory splits. The height of a peak is a measure of the strength of the cluster division. The expected number of clusters is equal to the value of *R* if the value of *Q* is a peak.

5 Experiments

In this section, we evaluate the algorithm NK-means using the well-known Iris dataset[6]. The performance of NK-means is compared with traditional K-means algorithm. Iris dataset includes 150 records, each record has four character attributes and one class attribute. The 150 records are partitioned into three classes on average according to the class attribute. In the processing of experiment, we calculate the similarity between nodes using angle-off cosine methods and set K=3. The result of the experiments is shown in Fig. 1. and Fig. 2.

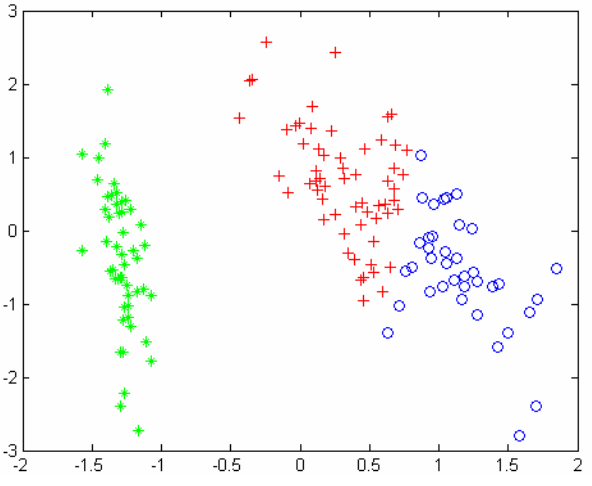


Fig. 1. The result by K-means

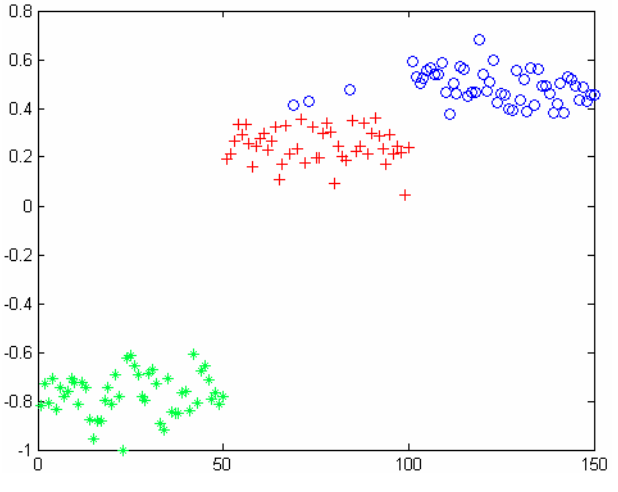


Fig. 2. The result by NK-means

In Fig. 1. and Fig. 2., the objects belonged to same cluster are shown with same color and same shape (such as '*', '+', ' o '), of which the distribution is more regular in shape and denser in structure in Fig. 2. than that of in Fig. 1., this verifies the conclusion: After the data are processed by Normal matrix, the influences of outliers are reduced and the clusters structure formed by the objects is sharper than ever before. Because class 1 (colored green in Fig. 1. and Fig. 2.) is far from the other two classes itself, the result of clustering is equal in K-means and in NK-means, However, the boundary between class 2 and class 3 in the original dataset is fuzzy, shown in Fig1., the result of K-means clustering leads to some of errors, the accuracy is 88.67%.

But class 1 and class 2 composed by the processed data using Normal matrix have sharp boundary, shown in Fig 2., the correct rate of NK-means algorithm up to 98%. The analysis of the experiments listed in Table 1.

Table 1. Algorithm's result on Iris dataset

Algorithm	K-means			NK-means		
Class	1	2	3	1	2	3
The number of the correct nodes in class	50	61	39	50	47	53
The number of error nodes in class	0	14	3	0	0	3
Accuracy	88.67%			98%		

We apply with NK-means algorithm to divide Iris data set into K clusters by the different value of K (K=2,3,…,149), the plot of the modularity, Eq. (5), is shown in Fig. 3.. Fig. 3 only represents the situation that the number of clusters is less than 20, because the modularites corresponding to the number of clusters located in the interval 21～149 are closer to 0. As we can see from Fig. 3, the modularity has a single clear peak, at the point where Iris data set breaks into two clusters, the peak value is

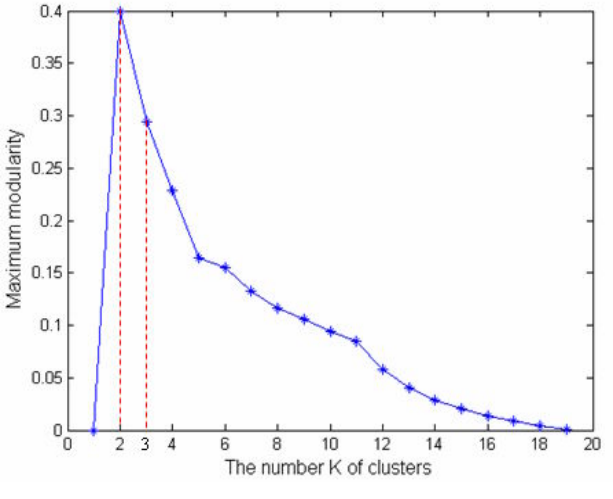


Fig. 3. The modularity for Iris by NK-means

0.40, that is K=2.. The peak in the modularity corresponding to a clear identification of the clusters shown in Fig. 2., which splits class 1 from class 2 and class 3. Furthermore, class 2 and class 3 are looked as a cluster because of the fuzzy boundary between them. From Fig. 3., we also see that the value of Q that is the closest to the peak is equal to 0.29, which corresponding to the number K=3 of clusters, it is a perfect identification of the clusters shown in Fig. 2.. So, the modularity defined in Eq. (5) could quantify the number K of clusters effectively, especially if the clusters structure is sharp.

6 Conclusions

In this paper, we proposed an improved K-means clustering algorithm, NK-means, which is based on spectral methods. NK-means make full use of the structural characteristics of Normal matrix to normalize the original dataset, and then finds clusters in the processed datasets by K-means algorithm. During the normalization, the data is transformed from a matrix into a vector, and the structures of the clusters are changed from three-dimensional sphere into two-dimensional circle. To select the number K of clusters, we also introduce a method of measure reflected the strength of the objects in the same cluster effectively. The experiment using well-known Iris dataset demonstrate an outstanding performance of the new method.

Future work will focus on the selection of the K centroids to optimize NK-means further.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2001)
2. Qian, W., Zhou, A.: Analyzing Polular Clustering Algorithms from Different Viewpoints. Journal of Software 13(8), 1382–1394 (2002)
3. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Bekeley (1967)
4. Wang, X., Li, X., Chen, G.: Theory and Application of Complex Network. Tsinghua Publishing House, Beijing (2006) (in Chinese)
5. Capocci, A., Servedio, V.D.P., Caldarelli, G., Colaiori, F.: Detecting Communities in Large Networks. Physica A 352(2-4), 669–676 (2005)
6. UCI Datasets from UCI, `http://www.sgi.com/tech/mlc/db/iris.all`

E-Means: An Evolutionary Clustering Algorithm

Wei Lu[1], Hengjian Tong[2], and Issa Traore[3]

[1] Faculty of Computer Science, University of New Brunswick, Fredericton Canada
[2] School of Computer Science, China University of Geosciences, China
[3] Department of Electrical and Computer Engineering, University of Victoria, Canada
wlu@unb.ca

Abstract. In this paper we propose a new evolutionary clustering algorithm named *E-means*. *E-means* is an *E*volutionary extension of k-*means* algorithm that is composed by a revised k-means algorithm and an evolutionary approach to Gaussian mixture model, which estimates automatically the number of clusters and the optimal mean for each cluster. More specifically, the proposed *E-means* algorithm defines an entropy-based fitness function, and three genetic operators for merging, mutation, and deletion components. We conduct two sets of experiments using a synthetic dataset and an existing benchmark to validate the proposed *E-means* algorithm. The results obtained in the first experiment show that the algorithm can estimate exactly the optimal number of clusters for a set of data. In the second experiment, we compute nine major clustering validity indices and compare the corresponding results with those obtained using four established clustering techniques, and found that our *E-means* algorithm achieves better clustering structures.

Keywords: Evolutionary Clustering, K-means, Evolutionary Computation, Gaussian Mixture Model, Clustering Validity.

1 Introduction

Clustering is the organization of data patterns into groups based on some measures of similarities. The main goal of clustering is grouping data objects in classes, so that intra-class similarity is maximized and inter-class similarity is minimized. Many clustering algorithms have been proposed in the last decades since there is no generic clustering algorithm that can be universally applicable in solving the variety of structures present in multidimensional data sets and each new clustering algorithm actually performs slightly better than existing ones in a fashion of specific distribution of patterns. Moreover, the completed knowledge of the domain and the data gathering process plays an important role in achieving higher quality feature extraction, similarity computation, grouping, and data abstraction.

K-means has been proved to be one of the most efficient clustering algorithms over years in terms of execution time compared to several other existing clustering algorithms. It has also been reported that k-means is one of the few algorithms that have been applied successfully to large data sets due to its linear and order-independent time and space complexities; most of the existing algorithms cannot handle large data

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 537–545, 2008.
© Springer-Verlag Berlin Heidelberg 2008

sets. There are, however, several weaknesses in k-means, which may represent serious impediments to both its effectiveness and efficiency for large-scale data clustering. The first significant limitation is that the k-means algorithm cannot process noisy data points or outliers, which play an important role in clustering analysis. Other two important shortcomings of k-means include cluster convergence, number of clusters dependency and degeneracy. The k-means algorithm may fail to converge to a global optimum if the initial partition is not properly chosen, a situation that occurs often and is not an obvious task. Clusters degeneracy stands for the fact that the clustering process may lead to empty clusters, which are useless for the classification process. The number of clusters dependency is related to the difficulty of identifying the optimal number *k* of clusters needed; this dependency is a serious problem. In order to explain this fact in more detail, we conduct a small case study based on a subset of 1999 KDDCUP dataset [1]. We selected two days data from the 1999 KDDCUP dataset. The first day includes only normal instances and the second day includes both anomalous and normal instances. The number of connection records in the first day is 4904 normal instances. The number of connection records in the second day is 4972 instances, in which 4904 are normal and 68 are anomalous. We apply k-means clustering algorithm to our two days data by assuming that the dataset includes exactly two clusters, normal and anomalous. Table 1 illustrates the results obtained with this approach.

Table 1. Preliminary clustering results using plain K-means algorithm

Number of detected anomalous instances		Number of normal instances detected as anomalous		False Positive Rate (%)	
1st day	2nd day	1st day	2nd day	1st day	2nd day
0	68	888	0	18.1	0.0

As illustrated in Table 1, when there is no anomalous instance in the dataset, the assumption that the number of clusters is 2 is invalid, and thus the k-means clustering system generates a false positive rate (FPR) of 18.1%, which is very high. In contrast, we found that when k-means clustering is applied to the same data based on a correct estimate of the number of clusters, the FPR decreased from 18.1 to 0%.

As indicated above, there are two important limitations of k-means affecting clustering effectiveness, namely determination of cluster number and detection of outliers. Determining the optimal number of clusters for a dataset is one of the most difficult issues in cluster analysis and is essential for effective and efficient data clustering. A popular clustering approach sensitive to this problem is based on Gaussian mixture model (GMM). GMM is based on the assumption that the data to be clustered are drawn from one of several Gaussian distributions and it was suggested that Gaussian mixture distribution could approximate any distribution up to arbitrary accuracy, as long as a sufficient number of components are used [5]. As a result, the entire data collection is seen as a mixture of several Gaussian distributions, and their corresponding probability density functions can be expressed as a weighted finite sum of Gaussian components with different parameters and mixing proportions [6]. A common approach for estimating the parameters of GMM is Expectation-Maximization (EM) algorithm [2]. Previous works on estimating the number of mixing components for the

GMM are mainly based on statistical techniques, which are prone to converge into local optima since they usually stop performing further search when the corresponding criteria reach certain thresholds. In this paper we tackle this issue of number of clusters estimation using a new evolutionary clustering approach, which combines the Gaussian mixture and the EM algorithm with the K-means algorithm. In contrast with previous statistical approaches, evolutionary computation schemes have inherently the potential capability to escape from local maximum since their search space for optimal solutions can be extended through evolutionary operations and optimization. Also, an important feature of the GMM is the capacity for outlier detection. Integrating evolutionary GMM with k-means results in a new evolutionary clustering algorithm, called E-means. Compared to traditional K-means, the E-means combines two distinct but complementary functionalities: (1) cluster number estimation and optimal selection of clusters' centers, using an evolutionary approach to mixture resolving, and (2) outlier detection and removal based on mixture resolving during each generation.

The rest of the paper is structured as follows. Section 2 presents the evolutionary approach to GMM. Section 3 is an outlier detection strategy. The detailed E-means algorithm is proposed in Section 4. Section 5 presents and discusses the results obtained in clustering validity, and Section 6 makes some concluding remarks.

2 An Evolutionary Approach to GMM

2.1 Representation of Individuals for Evolutionary Computation

A random variable x is said to follow a finite mixture distribution when its probability density function $p(x)$ can be represented as a weighted sum of kernels:

$$p(x) = \sum_{i=1}^{k} \alpha_i f_i(x; \mu_i, \sigma_i)$$

Where k is the number of mixture components, $f_i(x; \mu_i, \sigma_i)$ is the component kernel, μ_i is the mean of x, σ_i is the variance of x and $\alpha_i (1 \le i \le k)$ are the mixing proportions. In case each component is Gaussian, we say that variable x follows a Gaussian mixture model, in which kernel function $f_i(x; \mu_i, \sigma_i)$ can be a multivariate Gaussian or a univariate Gaussian. Each individual in the evolutionary population can be represented by a vector $< \alpha_i, \mu_i, \sigma_i > (1 \le i \le k)$. EM algorithm generates an estimate of the set of parameters $\{\alpha_i, \mu_i, \sigma_i\}(1 \le i \le k)$ and the posterior probabilities $p(i \mid x_n)(1 \le n \le N)$. The posterior probability describes the likelihood that the data pattern x_n approximates to a specified Gaussian component i. The greater the posterior probability for a data pattern x_n belonging to a specified Gaussian component i, the higher the approximation. As a result, each data x_n is assigned to the corresponding Gaussian component i according to $p(i \mid x_n)$ and final clustering results are statistically represented by the set of parameters $\{\alpha_i, \mu_i, \sigma_i\}$, which are also individuals.

2.2 Evolutionary Operators

During the evolution, we randomly merge, mutate or delete components. Therefore we have three evolutionary operators called *merging*, *mutation* and *deletion* operators, indicating for merging, mutating and removing the components, respectively.

Merging Operator: The merging operation combines two components leading to a new component, and thus decreasing the total number of components by 1. For univariate Gaussian components, the means, variances and mixing proportions for the new component are as follows:

$$\mu_{merge} = \frac{\alpha_1\mu_1 + \alpha_2\mu_2}{\alpha_1 + \alpha_2},\ \alpha_{merge} = \alpha_1 + \alpha_2$$

$$v^2_{merge} = \frac{\alpha_1}{\alpha_1 + \alpha_2}[v_1 + (\mu_1 - \mu_{merge})]^2 + \frac{\alpha_2}{\alpha_1 + \alpha_2}[v_2 + (\mu_2 - \mu_{merge})]^2$$

Otherwise, for the multivariate Gaussian component, all means, covariance matrixes, and mixing proportions are re-computed by the EM algorithm according to the new number of clusters.

Mutation Operator: The mutation operation mutates the mean and variance of a selected component with a random value, while keeping the same value of mixing proportions.

Deletion Operator: The delete operation eliminates a component entirely from the total number of components. After applying this operator, all other components update their mixing proportions to sum one and also keep corresponding means and covariance matrices unchanged.

2.3 Fitness Function

We define an entropy-based fitness function as the criterion for measuring how well the mixture model approximates the unknown density underlying the data samples: *the higher the fitness function value*, *the better the approximation*. The entropy-based fitness function EF_T is defined as follows:

$$EF_T = \sum_{i=1}^{k} \alpha_i \left| ef_i \right|$$

Where k is the number of mixture components and α_i $(1 \le i \le k)$ are the mixing proportions; ef_i denotes the individual fitness for each of the mixture component, which is defined as follows:

$$ef_i = \frac{H(C_i)}{H_{\max}(C_i)}$$

Where C_i stands for the set of data whose probabilities of belonging to the i^{th} component are the highest compared to other components. $H(C_i)$ represents the entropy of these data calculated as follows:

$$H(C_i) = -\sum_{x \in C_i} p(x) \log_2 p(x)$$

According to results from information theory, the theoretical maximum entropy for an individual component, denoted by $H_{max}(C_i)$, can be calculated as follows:

$$H_{\max}(C_i) = \frac{1}{2}\log((2\pi e)^n |\Sigma|)$$

Where Σ is the covariance matrix of C_i and n is the number of dimensions for data cluster C_i. It is established in information theory that Gaussian variables have maximum entropy among all the variables with equal variance. Hence, a high value of the individual entropy value ef_i for the i^{th} kernel corresponds to high probability that this component is truly gaussian and also represents a better match between probability distribution function and data. As a result, the higher the entropy-based fitness value EF_T, the better the approximation of the entire dataset.

3 Outlier Detection Based on GMM

GMM assumes that data patterns in a dataset can be drawn from one of several distributions. As a result, the entire data collection is seen as a mixture of several Gaussian distributions. However, it is possible that few data patterns cannot be categorized into none of the components of the mixture distribution. These data patterns are then naturally seen as the outliers for the dataset. Algorithm 1 illustrates the detailed outlier detection algorithm.

Algorithm 1. The proposed outlier detection algorithm

Function: GMM_Outlier_Detection (dataset and k) **returns** outlier data set
Inputs: dataset $X \sim \{x_n \mid n = 1,2,...,N\}$, and the number of components k
Output: outlier data set
Initialization:
outlier data set = ϕ; $j \leftarrow 0$; Initial parameters $\{\alpha_i^j, \mu_i^j, v_i^j\}$, $1 \le i \le k$, are randomly generated; Calculate the initial log-likelihood L_j;
Repeat: For $1 \le i \le k$, $1 \le n \le N$
If ($\alpha_i^j \ge outlier_{thres}$) **then** compute posterior probability $p_j(i \mid x_n)$;
Else $p_j(i \mid x_n) = 0$;
$j \leftarrow j + 1$;
Re-estimate $\{\alpha_i^j, \mu_i^j, v_i^j\}$ by using $p_{j-1}(i \mid x_n)$, $1 \le i \le k$, $1 \le n \le N$;
Calculate the current log-likelihood L_j;
Until: $|L_j - L_{j-1}| < th_1$ or $j > th_2$
For $1 \le i \le k$, $1 \le n \le N$
If ($p_{j-1}(i \mid x_n) = 0$), assign x_n to Outlier Data Set
Return Outlier Data Set;

In Algorithm 1, thresholds th_1 and th_2 correspond to the termination conditions associated with the outlier detection algorithm: th_1 is a measure of the absolute precision required by the algorithm and th_2 is the maximum number of iterations required by the algorithm. $outlier_{thres}$ refers to the minimum mixing proportion.

4 E-Means Algorithm

Algorithm 2 depicts the E-means algorithm. The algorithm combines the evolutionary approach to mixture modeling, the outlier detection strategy, and k-means clustering. During each generation, the evolutionary approach generates an optimal estimate of the number of clusters, the cluster centers and corresponding clusters by k-means. At the same time, the outlier detection based on GMM is used to find and remove the noisy data from the original dataset before the data are processed by k-means.

Algorithm 2. The E-means algorithm

Function E-means (data) **returns** clusters
Inputs: Collection of data samples
Initialization:
Randomly generate an initial population $population_0$ with initial number of components k; $j \leftarrow 0$; $population_j \leftarrow population$;
$population_{optimal} \leftarrow EM(population_j)$; $EF_{Toptimal} \leftarrow EF_T^j$;
Repeat: $j \leftarrow j+1$;
Apply genetic operators to $population_{optimal}$ yielding $population_j$;
$population_j \leftarrow EM(population_j)$; Compute EF_T^j ;
If $\left(EF_T^j > EF_{Toptimal}\right)$
do $population_{optimal} \leftarrow population_j$, $EF_{Toptimal} \leftarrow EF_T^j$;
$OutlierData = GMM_Outlier_Detection(population_{optimal}, k_{optimal})$
Remove outlier data from the original dataset and
$q \leftarrow 0$, initialize data clustering based on $population_{optimal}$: the number of clusters corresponds to $k_{optimal}$, and the means of each component in $population_{optimal}$ corresponds to the initial cluster centers c_i , $1 \le i \le k_{optimal}$
Repeat: $q \leftarrow q+1$
Assign data samples to clusters by determining the closest cluster center points. Calculate, the new center point c_{inew} for each cluster i.
Until: $\left|c_{inew} - c_i\right| < th_3$ or $q > th_4$
until: $EF_{Toptimal} > th_5$ or $j \ge th_6$ or $k = 0$
Return the clusters obtained

Sub_Function: EM (population) **returns** a population
Initialization: $r \leftarrow 0$; $population_r \leftarrow population$; Compute the log-likelihood L_r
Repeat: $r \leftarrow r+1$; Generate a new group of components $\{(\alpha_i^r, \mu_i^r, \sum_i^r) \mid 1 \le i \le k\}$ using E-step and M-step; $population_r \leftarrow \{<\alpha_1^r, \mu_1^r, \sum_1^r>, .., <\alpha_k^r, \mu_k^r, \sum_k^r>\}$; L_r
Until: $\left|L_r - L_{r-1}\right| < th_7$ or $r > th_8$
Return $population_r$

The algorithm starts with an initial number of components, which is randomly selected based on the size of the data. Suppose the total number of data objects is N, then the initial number of clusters is a random number in the range [1, N]. A population denoted by $population_j$ corresponds to a set of new individuals created during the j^{th} generation. We denote by k_j the size of $population_j$; i represent the i^{th} component generated ($1 \le i \le k_j$); j stands for the number of generations. We use $population_{optimal}$ and $k_{optimal}$ to denote the optimal individuals and the optimal number of clusters, respectively. New individuals are composed by $population_j$ and $population_{optimal}$. EF_T^j denotes the entropy-based fitness value associated with $population_j$, while $EF_{Toptimal}$ refers to the current optimal fitness value during evolution. The algorithm combines iteratively EM steps with evolutionary operations. EM is first used to estimate the best parameters for an initial number of clusters. The evolutionary algorithm is then used to estimate the best number of clusters given the (best) parameters obtained after the EM steps. The outcome of the evolutionary algorithm is re-submitted to EM algorithm, and the same two-stage process is repeated until the termination conditions are satisfied. During this loop, the outliers are moved to the outlier dataset from the original dataset and k-means is operated based on the current number of clusters and corresponding centers. The termination conditions occur when the generated number of new clusters becomes 0 or specified maximum entropy fitness value or numbers of iterations are reached.

Thresholds th_3 and th_4 correspond to the termination conditions for k-means algorithm: th_3 is a measure of the absolute precision required by k-means. The k-means algorithm terminates when the variation in distance between central points in two consecutive steps of the algorithm are less than threshold th_3. Threshold th_4 is the maximum number of iterations. Thresholds th_5 and th_6 correspond to the termination conditions associated with the evolutionary algorithm: th_5 is the maximum entropy fitness value and th_6 is the maximum number of iterations for the evolutionary algorithm. Thresholds th_7 and th_8 correspond to the termination conditions associated with the EM algorithm: th_7 is a measure of the absolute precision required by the algorithm. If the variation in log likelihood between two consecutive steps of the EM algorithm is less than this value, then the algorithm will terminate. Threshold th_8 is the maximum number of iterations of EM.

The time complexity of the proposed E-means algorithm is dependent on three main stages of execution, namely initial EM phase, evolutionary computation with recursive EM execution and plain k-means computation. In EM phase, the algorithm runs iteratively until the termination conditions are satisfied. The bottleneck is in the M step, where the posterior probabilities $p(i|x_n)$ are calculated. The time complexity of matrix inversion, matrix determinant, and matrix decomposition is $O(d^3)$, where d is the dimensionality of the data space. Besides, the time complexity of computing $p(i|x_n)$ is $O(d^2)$ for each pair of x_n and i. So, for all data objects and all components in GMM, the total time complexity of EM clustering is $O(i_{em}knd^2+i_{em}kd^3)$, where k is the number of components, n is the number of data objects and i_{em} is the number of iterations of EM algorithm. During the evolutionary computation, EM and K-means are recursively executed in each generation until the fitness function reaches its optimal value or the maximum number of generations is reached. During the plain k-means phase, we find that most of the time is spent on computing vector distances and one such operation costs $O(d)$. The traditional plain k-means algorithm involves two main

steps: a reassignment step and a recomputation step. During the reassignment step, each object is assigned to the group that has the closest centroid distance. In the reassignment step the time complexity for computing distances is $O(kn)$, so the overall complexity of this step is $O(knd)$. During the recomputation step, since all objects have been reassigned, the position of each cluster's centroid is recalculated. In the recomputation step, each data item is analyzed in order to compute the new clusters center points; so the complexity of this step is $O(nd)$. As a result, the overall complexity of the plain k-means stage is $O(iknd)$, where i is number of iterations. Since the number of clusters in each generation might be changed, we denote k' as the estimated maximum number of clusters. The time complexity of the evolutionary phase is $O(i_{ea}i_{em}k'nd^2 + i_{ea}i_{em}k'd^3 + i_{ea}i_{km}k'nd)$, where i_{ea} stands for the number of generations in the evolutionary computation scheme. Considering that the time complexity in evolutionary computation is much higher than the other two stages, we can conclude that the final time complexity of the E-means algorithm is $O(i_{ea}i_{em}knd^2 + i_{ea}i_{em}kd^3)$. The algorithm has linear time complexity in terms of either of the following parameters: size of the data (n), number of iterations for the evolutionary phase (i_{ea}), or number of iterations for the EM phase (i_{em}). It has cubic time complexity in terms of the dimension of the feature space (d). So, we can increase the size of the data without significantly impacting the running time of the algorithm. We, however, need to significantly decrease the dimension of the feature space in order to improve running performance.

5 Experimental Evaluation

During the evaluation, we conduct two sets of experiments with three different datasets to validate the E-means clustering algorithm. In the first experiment, we investigate the relationship between the probability of genetic operation and the algorithm's convergence speed. In the second experiment, we compare E-means with other four existing well-known clustering algorithms. The first dataset is a synthetic dataset, which contains 5000 data items. The data items were independently generated by 10 bi-dimensional normal distributions. The second dataset is the Pendigit dataset from [1], which contains handwritten digit images of the first 10 digits. We extract 100 data objects based on the first 5 digits (i.e. 0, 1, 2, 3 and 4). Each object is described using 16 features. Hence, the extracted dataset includes 5 clusters consisting of 16-dimensional data points.

During the first experiment, we studied 10 different sets of genetic operation probabilities, with merging probabilities varying from 0.3 to 0.99 and mutation probabilities ranging from 0.699 to 0.009. For each set of genetic operation probability, we run the clustering algorithm 10 times during the test. The maximum number of iterations was set to 500 and the fitness threshold was set to 0.98. During the evolution, the initial number of clusters is empirically set to 50. For synthetic dataset, the algorithm converges to the global optimum for all the test runs with merging probability equal or greater to 0.80, by identifying the exact number of clusters (i.e., 10) contained in the test dataset. With Pendigit dataset, the algorithm converges to the global optimum for all the test runs for merging probability equal or above 0.65. The exact number of clusters identified by our algorithm is 5 with Pendigit dataset. More detailed results about the relationship of coverage speed and number iterations can be found in [3].

The second experiment is a comparison of E-means with other four existing clustering algorithms (i.e. k-means, average linkage hierarchical algorithm, weak clustering and ensemble clustering) in terms of three clustering validity indices: *Dunn*, *Davis-Bouldin*, *Rand*, *Goodman-Kruskal*, *Silhouettes*, *Isolation*, *Jaccard*, and *ClassAccuracy* [4]. For instances, *Dunn* index is used to study cluster compactness and separateness. The larger the value of *Dunn*, the better the clustering result; a small value of *Davis-Bouldin* index indicates that the clusters are compact and far from each other; and *Rand* index measures the similarity between the resulting clustering structure *C* and the defined partition of the original data *P*. The higher the *Rand* value, the more similar *C* and *P* are. During the experiment, the merging probability and mutation probability is set as 0.95 and 0.03 respectively and clustering results show that our E-means algorithm achieves a maximum inter-cluster distance, a minimum intra-cluster distance, and the obtained clustering structure is more similar with the original dataset. Moreover, clustering results from E-means algorithm show that the obtained intra-clusters are compact and inter-clusters are far from each other. More detailed comparison results about the values of different validity indices and algorithm complexities can be found in [3].

6 Conclusions

We propose in this chapter a new hybrid clustering algorithm, named E-means. E-means algorithm improves the performance of the classical k-means algorithm by optimally adjusting the pre-determined parameters. E-means is composed by an evolutionary approach to mixture resolving and a revised k-means algorithm. The evolutionary approach is used to search the optimal number of clusters, and generate the center points for initial clusters. Empirical validations with synthetic data as well as real dataset show that the E-means algorithm can not only estimate exactly the global optimal number of clusters after few generations but also generates the optimal clustering structure.

References

1. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases, University of California, Irvine, Department of Information and Computer Sciences (1998)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data Via the EM Algorithm (with discussion). Journal of the Royal Statistical Society B 39, 1–38 (1977)
3. ISICA08 Experimental Results, `http://www.ece.uvic.ca/~wlu/ISICA08.htm`
4. Machaon CVE, `http://machaon.karanagai.com/`
5. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
6. Titterington, D., Smith, A., Makov, U.: Statistical Analysis of Finite Mixture Distributions. John Wiley & Sons, New York (1985)

Finding Motifs of Financial Data Streams in Real Time

Tao Jiang[1,2], Yucai Feng[1], Bin Zhang[2], Jie Shi[1], and Yuanzhen Wang[1]

[1] College of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, China
[2] Department of Computer Science,
Hengyang Normal University,
Hengyang 421008, China
jiangtao_guido@yahoo.com.cn

Abstract. Finding motifs of financial data streams in real time is a very interesting and valuable work. We hope to find the motif existing in financial data streams on local trend subsequence. A stock market trader might use such a tool to spot arbitrage opportunities or escape the underlying venture. The paper introduces a novel distance measurement, that is SDD (Slope Duration Distance), for local subsequences. At the same time, we propose an efficient algorithm of motif discovery over a great deal of financial data streams, that is PMDGS (P-**M**otif **D**iscovery based on **G**rid **S**tructure), which make use of PLA (Piecewise Linear Approximation) technology and grid structure. Extensive experiments on synthetic data and real world financial trading data show that our model provides several orders of magnitude performance improvement relative to traditional naive linear scan techniques.

1 Introduction

The applications of many domains are consisted of data streams, i.e., financial market, sensor network, etc. It is very interesting and valuable to find motifs of financial data stream. However, it is not easy to process such data in high speed data stream environment. This is because that stream time series have their own characteristics, compared to traditional achieved data: (1) Data are frequently updated in stream time series. Thus, previous approaches applied to traditional archived data may not work in the scenario. (2) Owing to the frequent updates, it is impossible to store all the data in memory or on disk, thus, efficient and one-pass algorithms are very important to achieve a real time response.

However, how to detect such motif among thousands of financial data streams? A kind of technique is subsequence similarity matching. It uses L_p-norm [1] or Dynamic Time Warping (DTW) [2] to metric the distance of two stream time series. When using L_p-norm, it requires two sub sequences to keep the same length. But, the actual subsequences generally have a different length. In addition, it is well-known that Euclidean measure is very sensitive to distortion

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 546–555, 2008.
© Springer-Verlag Berlin Heidelberg 2008

and noise [3]. However, direct DTW computation generally is time consuming and doesn't fit with stream environment in real time. Likewise, DTW also doesn't adapt to amplitude change.

There are some motif discovery approaches to help solve such problem, such as motif discovery in single time series [4], probabilistic motif discovery [5], shape motif discovery [6] and motif discovery under uniform scaling [7]. These approaches make use of different solved means to find motif or take consideration of motif discovery from different prospective. However, they work in a static environment and are generally not suitable for streams. On the other hand, they don't take consideration of financial stream characteristics.

The goal of motif discovery is detecting previously unknown, frequently occurring patterns from a knowledge discovery viewpoint. However, it is quite challenging to design an efficient algorithm in a streaming environment, especially in the face of a large of data streams. Formally, given n data streams with an average length m for each compared local pattern, the time complexity will be $O(mn^2)$ at each time tick t if we adopt traditional sequential scan compared method. Obviously, when n is larger, the scan method will not acquire a good efficiency on storage space and response time. Motivated by the random projection of above-mentioned approaches [5,6,7], we make use of a grid structure which is similar with the collision matrix to project the local pattern into it. Then, the grid unit which contains most projected local pattern points is choose. The corresponding local pattern is the search motif. So, our approach is fast. It only need $O(km)$ $(k \ll n)$ search time with a little of indexing storage.

2 Background and Basic Concepts

2.1 A Review of PLA

In our scheme, we use PLA (Piecewise Linear Approximation) [8] to divide the stream time series S into a sequence of subsequence $subS_i$ and approximate each subsequence $subS_i$ with the corresponding line segment L_i according to a certain accuracy threshold σ, where i is the sequence number and σ is a user-defined threshold, i.e., $\sigma = 10$. We call $subS_i$ local pattern lp_i. Formally, for a given sequence segment $S = (s_1, s_2, \ldots, s_n)$ of length n, PLA can use one line segment, $y_t = a \cdot t + b$ $(t \in [1, n])$, to approximate S, where a and b are two coefficients in a linear function such that the *approximation error*, $Err_{PLA}(S)$, of S is minimized. $Err_{PLA}(S)$ is defined as the squared Euclidean distance between the approximated and actual time series segment.

2.2 Basic Concepts

In above, a local pattern lp_i must satisfy the condition of $Err_{PLA}(lp_i) \leq \sigma$. In fact, the method is not precise because the price of different financial stream time series generally has a very large difference, for example Intel stock price is 23.8$ in August 1, 2007, however IBM stock price is 112.04$. So, we use the improved condition of $Err_{PLA}(lp_i)/\sum_{j=1}^{n}(s_j) \leq \delta$, where δ is a threshold,

i.e., $\delta = 0.01$. In practice, there are some short segment after segmenting. So we add a pruning process to handle such situation.

For each local pattern $subS_i$ of length n, we call the data items of it's two end, that is (s_1,1) and (s_n,n), *End Point*. At the same time, the local pattern ($=(s_1, \ldots, s_i)$) is called *semi-local pattern* where $SLen_{min} \leq i < n$, $SLen_{min}$ denotes the shortest length of semi-local pattern, for example $SLen_{min} = 4$. The local pattern ($=(s_1, \ldots, s_n)$) is called *whole local pattern*. A local pattern includes three trends, that is UP trend, Down trend and No trend.

To compare the similarity of two local pattern, we give the definition of their distance as follows.

Definition 1. *(Slope Duration Distance) Given any two local pattern objects lp_1 and lp_2 whose local trend are the same, $SDD(lp_1, lp_2) = \lambda_1.|k_1 - k_2|/(k_1 + k_2) + \lambda_2.|t_1 - t_2|/(t_1 + t_2)$, where k_i denotes the slope of lp_i when using PLA to approximate lp_i, and t_i ($t_i \in N$) denotes the number of data items of lp_i, that is time duration of lp_i, λ_1 and λ_2 ($0 < \lambda_1, \lambda_2 < 1$ and $\lambda_1 + \lambda_2 = 1$) are user-defined parameters. (k_i, t_i) are called local pattern feature point of lp_i.*

Definition 2. *(Local Pattern Match) For two local pattern objects lp_1 and lp_2 from different streams, if their distance satisfy the equality $SDD(lp_1, lp_2) \leq \epsilon$, we call lp_1 is similar with lp_2, ϵ is user-defined threshold.*

Definition 3. *Self Match: Given two semi-local pattern lp_1 and lp_2 if they come from the same data streams, we call lp_1 is self match with lp_2.*

We can now define the problem of enumerating the P most significant motifs among n data stream time series.

Definition 4. *P-Motifs: Given a set of n financial data streams TS which contains m ($1 \leq m < n$) semi-local pattern at time tick t and a threshold of local pattern similarity ϵ, the most significant motif among TS (called thereafter 1-Motif) is the subsequence C_1 that has the highest count of non-self matches. The P^{th} most significant motif among TS (called thereafter P-Motif) is the subsequence C_P that has the highest count of non-self matches, and satisfies $D(C_P, C_i) > 2\epsilon$, for all $1 \leq i < P$.*

Note that this definition forces the set of subsequences in each motif to be mutually exclusive. This is important because otherwise two motifs might share the majority of their elements, and thus be essentially the same.

Next, we show that distance function SDD is a metric function. This property is very important because only if it is satisfied, we use the triangular inequality filtering and fast indexing mechanism. Firstly, we give two lemmas as follows.

Lemma 1. *if $a, b, c \geq 0$, then the following inequality is true:*

$$\frac{|a-c|}{(a+c)} \leq \frac{|a-b|}{(a+b)} + \frac{|b-c|}{(b+c)} \tag{1}$$

Lemma 2. *if* $a, b, X_1, X_2, Y_1, Y_2 \geq 0$, $X_1 \leq X_2$ *and* $Y_1 \leq Y_2$, *then* $a.X_1 + b.Y_1 \leq a.X_2 + b.Y_2$.

Theorem 1. *Given two local pattern objects* lp_1 *and* lp_2, *their distance SDD* (lp_1, lp_2) *is metric.*

Proof. To prove $SDD(lp_1, lp_2)$ is metric, we need to prove it is symmetric and reflexive, and it satisfies the triangle inequality. It is obvious that $SDD(lp_1, lp_2) = SDD(lp_2, lp_1) \geq 0$ and $SDD(lp, lp) = 0$, so $SDD(lp_1, lp_2)$ is symmetric and reflexive. Next, we will prove that $SDD(lp_1, lp_2)$ satisfies the triangle inequality, i.e.,

$$SDD(lp_1, lp_3) \leq SDD(lp_1, lp_2) + SDD(lp_2, lp_3) \tag{2}$$

By definition 1, it is obvious that the distance of two local patterns are the summarization of a slope component and a duration component. If we prove (2) for both components, it will certainly be true for SDD by Lemma 2. Through Lemma 1, we have the following two equalities, that is, $\frac{|k_1-k_3|}{k_1+k_3} \leq \frac{|k_1-k_2|}{k_1+k_2} + \frac{|k_2-k_3|}{k_2+k_3}$ and $\frac{|t_1-t_3|}{t_1+t_3} \leq \frac{|t_1-t_2|}{t_1+t_2} + \frac{|t_2-t_3|}{t_2+t_3}$ is correct. In other words, inequality (2) is correct on each component of Theorem 1. So Theorem 1 is true.

3 Our Proposed Approach

3.1 Grid Structure Partition

The key idea of our approach is designing a good algorithm based on grid structure to improve the matching efficiency. In fact, the goal depends on the following factors, that is (i) how to partition the grid structure, (ii) how to update the grid structure, and (iii) a perfect searching algorithm.

Firstly, let's discuss the partition of grid structure. For this problem, generally, we need to consider two factors, that is, each interval magnitude and interval change (for example, equal interval or other interval), for the partitioning policy. However, in any case, we should try one's best to guarantee that matching local patterns are mapped into the same grid unit, otherwise different grid units. Before partitioning the grid structure, we firstly give a theorem as follows.

Theorem 2. *Given the distance threshold* ϵ *of SDD and a local pattern feature point* $A = (k_a, t_a)$, *if the grid structure is partitioned by the equal interval of the feature of k and t, for any point P of the rectangle which is the region formed by extending A leftward and downward* $\frac{2\epsilon}{1+\epsilon}$, *rightward and upward* $\frac{2\epsilon}{1-\epsilon}$, *respectively,* $SDD(P, A) \leq \epsilon$ *is true.*

Proof. Firstly, let's consider the case: $k_p \leq k_a$, by the condition, we have $k_a - k_p \leq \frac{2\epsilon}{1+\epsilon} k_a \Rightarrow k_a - k_p \leq (k_a + k_p)\epsilon$, so we have $\frac{|k_a-k_p|}{k_a+k_p} \leq \epsilon$. Next considering the case: $k_p \geq k_a$, by the condition, we have $k_p - k_a \leq \frac{2\epsilon}{1-\epsilon} k_a \Rightarrow k_p - k_a \leq (k_a + k_p)\epsilon$, so we have $\frac{|k_a-k_p|}{k_a+k_p} \leq \epsilon$. Synthesize the case $k_p \leq k_a$ and $k_p \geq k_a \Rightarrow \frac{|k_a-k_p|}{k_a+k_p} \leq \epsilon$. Multiplied λ_1 at its two sides, we have $\lambda_1 \frac{|k_a-k_p|}{k_a+k_p} \leq \lambda_1 \epsilon$ (denotes

as EQA). By a similar procedure, we have $\frac{|t_a - t_p|}{t_a + t_p} \leq \epsilon$. Multiplied λ_2 at its two sides, we obtain $\lambda_2 \frac{|t_a - t_p|}{t_a + t_p} \leq \lambda_2 \epsilon$ (denotes as EQB). From EQA and EQB, $\Rightarrow \lambda_1 \frac{|k_a - k_p|}{k_a + k_p} + \lambda_2 \frac{|t_a - t_p|}{t_a + t_p} \leq \lambda_1 \epsilon + \lambda_2 \epsilon$. $\because \lambda_1 + \lambda_2 = 1, \therefore \lambda_1 \frac{|k_a - k_p|}{k_a + k_p} + \lambda_2 \frac{|t_a - t_p|}{t_a + t_p} \leq \epsilon$.

For the searching region Rec of theorem 2, a visual illumination about it is show in Fig. 1. From the graph, we can observe the searching range of the feature point A which is located in grid unit $cell(4, 4)$ is $[k_a - \frac{2\epsilon}{1+\epsilon} k_a, k_a + \frac{2\epsilon}{1-\epsilon} k_a]$ and $[t_a - \frac{2\epsilon}{1+\epsilon} t_a, t_a + \frac{2\epsilon}{1-\epsilon} t_a]$. In Rec_a, B, C, D, E and F are the candidate items.

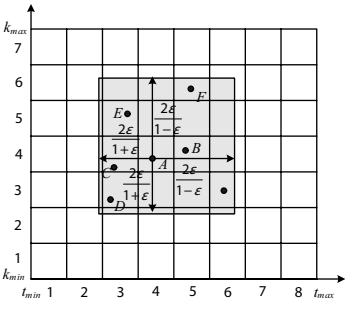


Fig. 1. The grid structure used to illuminate the searching range of feature point A

According to Theorem 2, the length of each grid unit need to gradually increases with the increase of k and t. In fact, the increase rate should be $(1 + \epsilon)/(1 - \epsilon)$. So, in our scheme, we use the equation $k_{i+1} = k_i * (1 + \epsilon)/(1 - \epsilon)$ and the equation $t_{i+1} = t_i * (1 + \epsilon)/(1 - \epsilon)$ to partition grid structure, where k_i and t_i denote the start position of i-th grid unit, respectively. The actual magnitude has a little difference with the computing value using above equations, owing to the limitation of minimal unit. Now, we have solved the first problem. We will solve the second problem and the third problem in the next sections.

3.2 PMDGS Algorithm

After partitioning the grid structure, we give P-Motif discovery algorithm PMDGS as algorithm 1, which uses the data-driven mechanism, reflected at line 2-8. Line 3 builds the grid index (denote it as $index_{grid}$) which is consisted of a set of feature point lists of each grid unit. Figure 2 shows an example of $index_{grid}$ structure. Each grid unit corresponds to a hashed position of $index_{grid}$, for example $c(3, 2)$, which records the hashed address $hashAddr$ and the number of mapped local pattern points $lpCount$. The mapped feature point f $(=(k, t))$ is stored in the corresponding list whose each node can be described into a triangle tuple ($StreamID$, k, t). System decreases the time of motif discovery by sorting $lpCount$ of each grid unit in descending order. Then $Motif_Search$ algorithm is invoked at line 6. Line 7-8 of PMDGS algorithm updates the index $index_{grid}$ when a new $End\ Point\ EP_{new}$ is identified.

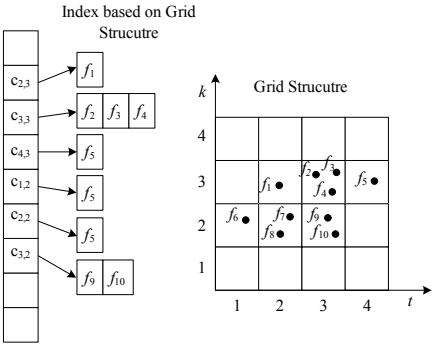


Fig. 2. A example to illuminate the grid index structure

Algorithm 1. PMDGS

Input: ϵ-the threshold of local pattern similarity, P-the threshold of P-th motif, $index_{grid}$-grid structure index
Output: the corresponding local patterns of k motif
1. Construct grid structure according to corresponding partition method;
2. **While** (a new data item (s_t, t) arrives)
3. Insert the semi-local pattern point into $index_{grid}$ by hashing method;
4. Sort the number of local pattern points contained in each grid unit in descending order;
5. Record the sorted result into array $sArray$, including hashed address and number of feature points of each grid unit;
6. Invoke *Motif_Search* to report P motif local patterns;
7. **If** (the data item (s_t, t) is an *End Point*)
8. Clear its all semi-local pattern points from the grid units of $index_{grid}$;

In *Motif_Search* algorithm, line 4-9 obtains the possible motif lp_{cur} of each grid unit and its location is recorded in $current_motif_location$. Then, *Motif_Search* looks for next motif which reflects at line 10-15, by comparing the distance of current motif lp_{cur} and all previous $motif_count_so_far$ motifs (denote previous i-th motif as lp^i_{pre}, $1 \leq i \leq motif_count_so_far$), which are stored in the motif set $\{P - motif\}$. If $SDD(lp^i_{pre}, lp_{cur}) \geq \epsilon$ is satisfied for each previous motif lp_{pre}, then the current motif lp_{cur} will be reported as discoverable $(motif_count_so_far + 1)^{th}$ motif . Otherwise, *Motif_Search* continues to search from the next grid unit until P motifs are discovered, reflecting at line 2. In fact, the theory proof of *Motif_Search* comes from the following theorem 3.

Theorem 3. *For each grid unit, it only possibly contains one motif if the grid unit is strictly partitioned according to theorem 2.*

Proof. We assume that there exist two motif lp_1 and lp_2 in a same grid unit $cell_i$. In term of theorem 2, it is obviously that the inequality $SDD(lp_1, lp_2) \leq \epsilon$

Algorithm 2. Motif_Search($sArray$, ε, P)

Input: $sArray$-the array stored *count* and *hashAddr* of each grid unit in order, ε-the threshold of local pattern match, P-P-th motif threshold.
Output: the corresponding local pattern of P motif

1. $motif_count_so_far = 0$, $P - motif = \Phi$; //initialization
2. **for** (i=1; $motif_count_so_far \leq P$; $i = i + 1$)
3. $max_point_count_so_far = 0$;
4. **for** (j=1; $j <= sArray[i].count$; $j = j + 1$)
5. Obtain searching rectangle Rec_j of current lp_j;
6. Store the *non-self match* number of lp_j contained in Rec_j into *count*;
7. **if** ($count > max_point_count_so_far$)
8. $max_point_count_so_far = count$;
9. $current_motif_location = j$; // denotes the local pattern as lp_{cur}
10. **if** ($motif_count_so_far == 0$)
11. $motif_count_so_far = 1$;
12. $\{P - motif\} \leftarrow lp_{cur} \cup \{P - motif\}$;
13. **else if** (each previous motif $lp^i_{pre} \in \{P - motif\}$ satisfies $SDD(lp^i_{pre}, lp_{cur}) \geq 2\epsilon$)//$1 \leq i \leq motif_count_so_far$
14. $motif_count_so_far = motif_count_so_far$+1;
15. $\{P - motif\} \leftarrow lp_{cur} \cup \{P - motif\}$;

is satisfied. Since $SDD(lp_1, lp_2) \leq \epsilon$, $SDD(lp_1, lp_2) > 2\epsilon$ will not satisfy. So, we have that theorem 3 is correct, in the light of definition 4.

3.3 Index and Update of Grid Structure

We use B^+-tree to index the corresponding data items. Our system indexes two type data. The first type is all data items ($=\{s_1, \ldots, s_{cur}\}$) of current local pattern lp which begin from the last identified *End Point* EP_{last} and end right before the current data point (s_{cur}, t_{cur}) from all data streams, so as to find the next new *End Point* EP_{new}. The second type is hashed address from the first *semi-local pattern* lp_{first} to current *semi-local pattern* lp_{cur} from all data streams, so as to update the grid structure when EP_{new} is identified.

Next, we introduce how to update the grid structure. We use a dynamic policy to maintain the grid structure. Whenever identifying EP_{new}, all corresponding *local pattern feature point* $\{(k_{first}, t_{first}), \ldots, (k_{last}, t_{last})\}$ of lp, formed by all *semi-local pattern*, is cleared from the corresponding grid unit of $index_{grid}$ by searching B^+-tree and obtaining hashed address list $hastAddr$ of lp.

3.4 Time and Space Complexity

Given N financial data streams and the average length of all local patterns m. The total space will be $N * (m - SLen_{min})$ *double* bytes and $N * (m - SLen_{min})$ *int* bytes with a little of extra index main memory. By a similar analysis, the space of B^+-tree is $N * m$ *double* bytes and $N * (m - SLen_{min})$ *int* bytes, in addition to the extra index space. Owing to $m \ll N$ when N is larger, so the actual space is linear, and much less than original size of dataset.

The most time-consuming part of our algorithm is $Motif_Search$ algorithm. Its time complexity is $O(P * FPNum_{gu})$, which is linear, where $FPNum_{gu}$ $(=N*(m-SLen_{min})/(Num_k * Num_t))$ denotes the average number of feature points contained in each grid unit, Num_k and Num_t is the interval number on k-axis and t-axis, respectively, and P represents the number of motifs.

4 Experimental Evaluation

4.1 Experiment Setup

In experiments, we use more than 1,820,000 data points from 1700 real stocks in China. We conducted experiments on a Pentium 3 PC with 512M memory. Some parameters are set up as follows. We use daily close price as financial data streams. The shortest length of local pattern $SLenmin$ is 4. The maximal relative approximation error δ is 0.01 on segmenting stream time series. The parameter ε, λ_1 and λ_2 are 0.1, 0.65 and 0.35, in pattern similarity, respectively.

4.2 Mining Financial Streams Motifs

In financial domain, finding motif by using computers to help the market judgement is also very interest. This is especially challenged when facing a large of financial data streams. To demonstrate the problem, we chose 1600 different financial stream time series and performed an experiment to find 1-Motif. Figure 3 gives a hint result which does not include all streams and only shows the stream time series of eight stocks. In the graph, we can distinctly observe that stream A, B, C, D, E and F are match from $t193$ to $t200$. In other words, there exists a 1-Motif during the period of $t193 - t200$ among all streams. In fact, this is affected by the national financial policy of the adjustment on required reserve ratio from 16.5% to 17.5% at June 7, 2008 in China.

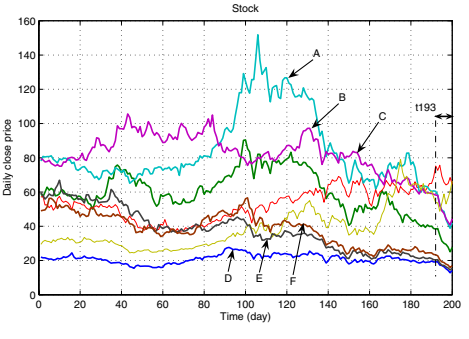


Fig. 3. A motif happens during t193-t200 (from June 6, 2008 to June 18, 2008) with $\epsilon = 0.1$. To illuminate the problem, eight representative stocks are used in the graph

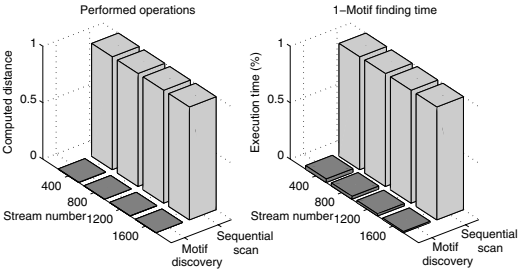


Fig. 4. Compared to SS by changing the number of streams

4.3 Efficiency of Motif Discovery

In previous subsection, we have shown that motif discovery algorithm is very effective in finding financial streams motifs. In this subsection, we will further demonstrate that our approach is not only effective but also efficient, which allows us to discover motifs in linear time with high precision.

We firstly compare two strategies for 1-motif discovery: *sequential scan* (denotes as SS) method and our *motif discovery* (denotes as MD) described in Section 3. SS performs an exhaustive search, computing SDD distance for each pair of stream time series. The running time of two strategies are given in Figure 4. We can see that MD algorithm prunes more than 99.9% distance computations of SS. For running time, MD spends time on three parts: computing k and t by PLA approximation, index maintaining time (insert, update, etc), and searching true motif from grid structure. The first part and the second part are almost constant. Actually the second part and the third part are most time-consuming. Overall, MD algorithm is 1 to 2 orders of magnitude faster than SS method which is clearly shown in right of Figure 4.

Next, we also compute the average speed improvement for motif discovery with different parameter P. Figure 5 left presents the improvement of distance computation in terms of the number of performed operations, as compared with the SS method. For all P, MD performs less than 1% of the operations

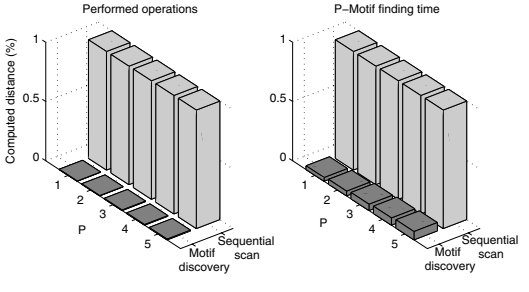


Fig. 5. Compared to SS with 1600 streams time series by changing P

performed by SS method. The speed-up introduced by MD is presented in Figure 5 right. The result shows that MD takes only 3%-7% of the time necessary for the SS method to complete when P is changed from 1 to 5.

5 Conclusion

We have formalized the problem of finding motifs among thousands of financial stream time series, and introduced an algorithm PMDGS to efficiently detect the motifs in real time. Experimental results show that our approach can efficiently find motifs with high precision. In the future, we plan to extend our approach to the other domains, i.e., network traffic data and astronomical data, etc.

References

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Lomet, D.B. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–74. Springer, Heidelberg (1993)
2. Berndt, D.J., Clifford, J.: Finding patterns in time series: a dynamic programming approach. In: Advances in Knowledge Discovery and Data Mining, pp. 229–248. AAAI Press, Menlo Park (1996)
3. Keogh, E.: Exact indexing of dynamic time warping. In: VLDB, pp. 406–417 (2002)
4. Patel, P., Keogh, E., Lin, J., Lonardi, S.: Mining Motifs in Massive Time Series Databases. In: ICDM, pp. 370–377 (2002)
5. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic Discovery of Time Series Motifs. In: SIGKDD 2003, pp. 493–498 (2003)
6. Xi, X., Keogh, E., Li, W., Mafra-neto, A.: Finding Motifs in a Database of Shapes. In: Jonker, W., Petković, M. (eds.) SDM 2007. LNCS, vol. 4721, pp. 249–260. Springer, Heidelberg (2007)
7. Yankov, D., Keogh, E., Medina, J., Chiu, B., Zordan, V.: Detecting Time Series Motifs Under Uniform Scaling. In: SIGKDD, pp. 844–853 (2007)
8. Chen, Q., Chen, L., Lian, X., et al.: Indexable PLA for Efficient Similarity Search. In: VLDB, pp. 435–446 (2007)

Improve the Accuracy of One Dependence Augmented Naive Bayes by Weighted Attribute

Siwei Jiang and Zhihua Cai

Faculty of Computer Science, China University of Geosciences, Wuhan 430074, China
amosonic@gmai.com, zhcai@cug.edu.cn

Abstract. Naive Bayes is a effective and widely used data mining algorithm for classification, but its unrealistic attribute conditional independence harm its performance. Selecting attributes subsets is an important approach to extend the Naive Bayes, and the state-of-the-art SBC algorithm has better accuracy in classification. In this paper, we review the weighted attribute method for Naive Bayes, and explain SBC is one of the special case in weighted attributed methods. Interesting this method, we present a new one dependence augmented Naive Bayes with weighted attribute called WODANB, which use the fuzzy Support Vector Machine to optimize the weights. Experiment on whole 36 datasets recommended by Weka, results show that WODANB significant outperforms than NB, SBC, ODANB, TAN.

Keywords: Naive Bayes, selective attribute, weighted attribute, support vector machine, classification.

1 Introduction

Classification is one of the fundamental issue in machine learning and data mining. Naive Bayes is one of the powerful and simply algorithm. Assume $A_i, i = 1, 2, .., n$ are n attributes which take values $a_i, i = 1, 2, .., n$ respectively, and C is the class label. An unlabelled instance $E = (a_1, a_2, \cdots, a_n)$ will be classified into the class C with the maximum posterior class probability $P(C|E)$.

In Naive Bayes classifier, all attributes are independent given the class:

$$P(C|E) = arg \max_c P(c) \prod_{i=1}^{n} P(a_i|c) \tag{1}$$

To avoid dive by zero, the probability in Naive Bayes is modified as follows, the training size is l:

$$P(c) = \frac{\sum_{i=1}^{l} \delta(c_i, c) + \frac{1}{n_c}}{n + 1} \tag{2}$$

$$P(a_i|c) = \frac{\sum_{i=1}^{l} \delta(a_{ij}, a_i)\delta(c_i, c) + \frac{1}{n_i}}{\sum_{i=1}^{l} \delta(c_i, c) + 1} \tag{3}$$

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 556–561, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Although Naive Bayes is a simple and effective classifier, the unrealistic independence assumption often makes its probability estimator incorrect. Selecting attributes subsets is an important direction to extend Naive Bayes, such as SBC[1], and selective attribute is a special case in Weighted Attribute method. Interesting in this research, we present a new algorithm One Dependence Augmented Naive Bayes with Weighted Attribute called WODANB, which optimize the weights of attributes by using fuzzy Support Vector Machine(FSVM)[5].

The rest of the paper is organized as follows. In section 2, we review some researches about NB. In section 3, the new algorithm WODANB is presented, it weights the attributes for ODANB and optimize the weighted parameters by FSVM. In section 4, Experiment on 36 benchmark datasets has show that WODANB significantly outperforms than NB, SBC, ODANB, TAN. In section 5, we make conclusions and discuss the future research on Naive Bayes.

2 Related Work

Selecting attributes subsets is one of the important directions to extend NB. Langley and Sage [1] presented an algorithm called selective Bayesian classifiers (simply SBC), which uses a forward backward greedy search method to select an attribute subset through the whole space of attributes.

Tree Augmented Naive Bayes(simply TAN) is a state-of-the-art algorithm, which construct the conditional dependence like the structure of tree, every attribute has only one parent attribute or no parent attribute. Ljiang [2] presented an algorithm called One Dependence Augmented Naive Bayes(simply ODANB). It select the attribute parent with maximal conditional mutual information, and delete the arc when the conditional mutual information is lower than average threshold.

$$P(C|E) = arg\max_{c} P(c)\prod_{i=1}^{n} P(a_i|a_{pi}, c) \tag{4}$$

We set the different weights for different attributes, which presents the attributes' contributions for the performance of Naive Bayes. The weighted attribute Naive Bayes(simply WNB) is defined as follows, where w_i is the weight for attribute a_i.

$$P(C|E) = arg\max_{c} P(c)\prod_{i=1}^{n} P(a_i|c)^{w_i} \tag{5}$$

when $w_i = \{0, 1\}$, the WNB transform to SNB. So, SNB is a special case of WNB. Hall [3] presents a decision tree-based attribute weighting algorithm, which estimates the degree of attribute dependency by constructing unpruned decision trees and looking at the depth at which attributes are tested in the tree. Gartner[4] use the support vector machine to evolute the weight attribute, it can solve the binary classes problem and the weight attribute is based on conditional independence.

3 ODANB with Weighted Attribute by Using Fuzzy Support Vector Machine

In Weighted attribute Naive Bayes, how to optimize the weight is a crucial issue. Support Vector Machine is a useful tool to parameter optimization. In binary classed problem, we defined the class is $class = \{True, False\}$.

$$\log \frac{P(T|E)}{P(F|E)} = \log \frac{P(T)\prod_{i=1}^{n} P(a_i|T)^{w_i}}{P(F)\prod_{i=1}^{n} P(a_i|F)^{w_i}} = \log \frac{P(T)}{P(F)} + \sum_{i=1}^{n} w_i log \frac{P(a_i|T)}{P(a_i|F)} \quad (6)$$

If it larger than 0, then the instance E belongs to class T, else the instance E belongs to F.

Support Vector Machine is a effective and elegant data mining tool which is base on structure risk minimization principle. SVM map the input space into the high dimension by kernel function, and get the separated hyperplane which maximize the margin between positive and negative points. The the decision function of SVM can be written as:

$$f(x) = sgn(W^T \phi(x) + b) = sgn(\sum_{i=1}^{l} \alpha_i y_i \phi(x_i)\phi(x) + b). \quad (7)$$

When the kernel function select as linear function $k(x_i, x_j) = \phi(x_i)\phi(x_j) = x_i^T x_j$, the decision function can be transformed as follow, where $x = \{a_1, \cdots, a_n\}$.:

$$f(x) = sgn(\sum_{i=1}^{n} w_i a_i + b). \quad (8)$$

From equation 6 and equation 8, we can transform the formulation WNB to SVM by the kernel function as follow:

$$\phi(x_i) = \{\log \frac{P(a_1|T)}{P(a_1|F)}, \cdots, \log \frac{P(a_n|T)}{P(a_n|F)}\} \quad (9)$$

Since weighted attribute is a powerful method for Naive Bayes, naturally we can suppose that whether it is useful for augmented Naive Bayes. Inciting by this propose, we present a new algorithm One Dependence Augmented Naive Bayes with weighted attribute(simply WODANB).

Ljiang[2] present a new algorithm ODANB. At first, ODANB select the parent of attribute which has the maximal conditional mutual information, then ODANB delete the arcs when their conditional mutual information below than threshold, because these arcs maybe overfit the training set and harm the performance of augmented Naive Bayes.

$$\begin{aligned} I_P(A_i; A_{Pi}|C) &= arg\max_j I_P(A_i; A_j|C) \geq threshold \\ threshold &= \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i=1}^{n} I_p(A_i; A_j|C) \end{aligned} \quad (10)$$

Using the equation 2, the probability is modified and it will not be zero, so the map function is always valid. Similar to equation 9, the map function of WODANB can be defined as:

$$\phi(x_i) = \{\log \frac{P(a_1|a_{p1}, T)}{P(a_1|a_{p1}, F)}, \cdots, \log \frac{P(a_n|a_{pn}, T)}{P(a_n|a_{pn}, F)}\} \tag{11}$$

For multi class problem, SVM construct $m*(m-1)/2$ binary SVMs, and decision the class label by voting from these pairwise classifiers, where m is the number of classes.

$$arg \max_{i=1,\cdots,m} f_i(x) = \sum_{j=1, j\neq i}^{m} f_{ij}(x) \tag{12}$$

Because Support Vector Machine for classification is based on the binary problem, when we adopt the vote strategy to judge the class label for multi-classes, there exist some unclassifiable regions. When instance lies in this region, we can adopt the fuzzy rule to classify it[4].

For multi-class problem, the member functions are the pairwise classifications:

$$\begin{aligned} m_i(x) &= \min_{j=1,\cdots,m, j\neq i} f_{ij}(x) \\ class &= arg \max_{i=1,\cdots,m} m_i(x) \end{aligned} \tag{13}$$

4 Experiment and Results

The experimental data are 36 UCI datasets recommended by Weka. All these datasets come from the UCI repository. To deal with the problems such as *MissingValues* and *numeric attributes*, in our experiment, we filter the datasets with three preprocessing stages by Weka system. They mainly include the following three processes:

1. We used the filter of *ReplaceMissingValues* in Weka to replace the missing values of attributes.
2. We used the filter of *Discretize* in Weka to discretize numeric attributes. Thus, all the attributes are treated as nominal.
3. It is well-known that, if the number of values of an attribute is almost equal to the number of instances in a data set, this attribute does not contribute any information to classification. So we use the filter of *Remove* in Weka to delete these attributes. In the 36 datasets, there only exists three this type of attributes, namely Hospital Number in data set horse-colic.ORIG, Instance Name in data set Splice and Animal in data set zoo.

we compare our algorithm with other algorithms via the ten-fold cross validation for all datasets. In all experiments, each algorithm above builders its *classifier* on the *training sets* of each datasets, then it uses the classifier to assign the class label to *test sets* of each datasets. The accuracy of each algorithm is evaluated

Table 1. Result of tow-tailed t-test on accuracy

	NB	SBC	ODANB	TAN
SBC	6/29/1			
ODANB	10/26/0	6/27/3		
TAN	10/23/3	6/26/4	7/27/2	
WODANB	11/25/0	9/27/0	7/29/0	7/29/0

Table 2. Experimental results on accuracy and standard deviation

Dataset	(NB)	(SBC)	(ODANB)	(TAN)	(WODANB)
anneal	94.32±2.38	96.88± 2.50 ◦	96.55± 1.11 ◦	98.77± 1.23 ◦	98.66± 1.15 ◦
anneal.ORIG	87.53±4.69	88.75± 3.72	90.31± 2.57	91.09± 3.66	92.76± 4.23
audiology	71.23±7.03	76.01± 7.05	62.73±10.19	77.33± 7.77	74.70± 8.58
autos	64.83±11.18	67.71±11.27	78.55± 6.92 ◦	79.48± 6.17 ◦	82.36± 8.76 ◦
balance-scale	91.36±1.38	91.36± 1.38	91.36± 1.38	86.09± 3.81 •	91.20± 1.35
breast-cancer	72.06±7.97	73.45± 8.91	69.61± 8.45	64.72± 7.07 •	70.31± 8.51
breast-w	97.28±1.84	96.42± 2.26	96.99± 1.85	95.27± 2.45	96.71± 1.92
colic	78.81±5.05	81.77± 4.89	81.25± 5.46	79.08± 7.08	79.35± 6.00
colic.ORIG	75.26±5.26	75.53± 6.15	68.76± 4.55	71.46± 5.79	69.32± 7.94
credit-a	84.78±4.28	85.51± 4.16	82.90± 3.54	82.17± 5.59	86.09± 6.08
credit-g	76.30±4.76	74.10± 3.87	73.40± 4.58	75.00± 5.73	74.10± 4.20
diabetes	75.40±5.85	75.53± 5.07	73.84± 7.31	74.60± 4.14	73.84± 6.53
glass	60.32±9.69	57.99± 6.89	60.28± 9.31	60.78± 9.33	62.58±10.94
heart-c	84.14±4.16	82.47± 7.61	80.46±10.31	75.88± 7.45 •	78.49±11.20
heart-h	84.05±6.69	79.00± 9.77	79.66± 5.97	78.95± 6.83	78.62± 6.04
heart-statlog	83.70±5.00	79.26± 9.75	80.00±11.07	76.30± 9.43	78.15±10.97
hepatitis	83.79±8.79	80.63± 6.80	85.13± 7.36	81.79± 9.73	81.29± 7.05
hypothyroid	92.79±1.02	93.53± 0.66	92.63± 0.82	93.16± 1.11	93.13± 0.56
ionosphere	90.89±3.49	91.17± 4.12	90.90± 5.10	93.75± 2.89	93.45± 3.81
iris	94.67±8.20	97.33± 4.66	94.67± 8.20	92.00± 7.57	96.00± 4.66
kr-vs-kp	87.89±1.81	94.34± 1.23 ◦	90.52± 1.54 ◦	93.55± 1.52 ◦	95.56± 1.53 ◦
labor	93.33±11.65	77.00±11.91 •	90.00±14.05	84.67±16.64	88.00±13.90
letter	70.00±0.81	70.57± 0.88 ◦	79.76± 0.88 ◦	81.21± 0.77 ◦	84.76± 0.68 ◦
lymph	85.67±9.55	79.00± 6.84	82.43± 7.18	80.43±11.05	81.71±10.12
mushroom	95.57±0.45	99.67± 0.23 ◦	99.94± 0.09 ◦	99.73± 0.08 ◦	100.00± 0.00 ◦
primary-tumor	46.89±4.32	46.02± 5.19	44.26± 4.06	46.01± 4.72	46.30± 5.44
segment	88.92±1.95	90.43± 1.96	94.20± 1.12 ◦	94.03± 1.67 ◦	95.11± 1.58 ◦
sick	96.74±0.53	97.59± 0.69 ◦	97.59± 0.48 ◦	97.56± 0.56 ◦	97.69± 0.56 ◦
sonar	77.50±11.99	70.71±12.97	77.02±11.28	76.05±14.46	70.69±10.14
soybean	92.08±2.34	91.79± 2.72	91.51± 3.94	95.75± 1.61 ◦	95.32± 2.15 ◦
splice	95.36±1.00	94.76± 1.60	93.98± 1.54	95.39± 1.35	96.33± 1.21
vehicle	61.82±3.54	60.65± 4.73	71.04± 2.80 ◦	71.88± 4.52 ◦	72.69± 3.84 ◦
vote	90.14±4.17	95.18± 3.93 ◦	94.04± 3.90 ◦	92.89± 4.48	97.01± 2.17 ◦
vowel	67.07±4.21	68.69± 3.47	91.82± 2.31 ◦	94.55± 1.73 ◦	95.15± 1.15 ◦
waveform-5000	79.96±1.92	81.32± 1.54	81.26± 0.91	79.18± 1.86	82.42± 1.55
zoo	94.18±6.60	93.18± 7.93	95.18± 8.15	95.09± 5.18	97.09± 6.65
Average	82.41	82.09	83.46	83.49	84.64

◦, • statistically significant improvement or degradation.

by the percentage of correct classifications on the test sets. Finally, we conduct two-tailed t-test with significantly different probability of 0.95 to compare our algorithm with other algorithms.

Table 1 and Table 2 show that our algorithm outperforms NB, SBC, ODANB, and TAN significantly. Now, we summarize the highlights as follows:

1. Our algorithm's performance measured by classification accuracy outperforms NB, SBC, ODANB, and TAN significantly. the *w/t/l* value between WODANB and NB is 12/25/1, between WODANB and SBC is 8/25/3,

between WODANB and ODANB is 10/23/3, between WODANB and TAN is 10/29/0.
2. The average classification is measured, WODANB average classification accuracy is 84.64, NB's average classification accuracy is 82.41, SBC's average classification accuracy is 82.09, ODANB's average classification accuracy is 83.46, TAN's average classification accuracy is 83.49.
3. SBC is a special case of weighted attribute Naive Bayes, SBC's weight is only equal to 0 or 1. We use the Fuzzy SVM algorithm to optimize the weight of ODANB, the weight can be real number. The $w/t/l$ value between WODANB and SBC is 9/27/0. So WODANB can get more precision value which represents the contribution of every attribute for the performance of Naive Bayes.

5 Conclusion

There are three approaches to extend the performance of Naive Bayes. In this paper, we explain that selective Naive Bayes is a special case of weighted Naive Bayes.

Interesting in the weighted attribute and augmented approaches, we adopt fuzzy Support Vector Machine to optimize the attributes' weight of ODANB (simply WODANB). Experiment on 36 datasets recommended by Weka, the results show that WODANB significantly outperforms than NB, SBC, ODANB, TAN. The future work will focus on weight attribute for different augmented Naive Bayes.

References

1. Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp. 339–406 (1994)
2. Jiang, L., Zhang, H., Cai, Z., Su, J.: One Dependence Augmented Naive Bayes. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS, vol. 3584, pp. 186–194. Springer, Heidelberg (2005)
3. Hall, M.: A decision tree-based attribute weighting filter for naive bayes. Knowledge-Based Systems 20, 120–126 (2007)
4. Gartner, T., Flach, P.A.: WBCsvm: Weighted Bayesian Classification based on Support Vector Machines. In: ICML 2001, pp. 156–161. Morgan Kaufmann, San Francisco (2001)
5. Abe, S., Inoue, T.: Fuzzy Support Vector Machines for Multiclass Problems. In: Proc. ESANN 2002, Bruges, Belgium, pp. 113–118 (April 2002)
6. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Multi-Relational Classification in Imbalanced Domains

Guangmei Xu[1], Hong Bao[1], and Xianyu Meng[2]

[1] Institute of Information Technology, Beijing Union University
100101 Beijing, China
xgm_xgm@126.com, baohongt@sina.com
[2] School of Computer Science & Engineering,
Liaoning University of Technology,
121001 Jinzhou, China
chinaxuguangmei@hotmail.com

Abstract. This paper discusses the problem of multi-relational classification on imbalanced datasets. To solve the class imbalance problem, a new multi-relational Naive Bayesian classifier named R-NB is proposed, the attribute filter criterion based on mutual information is upgraded to deal with multi-relational data directly and the basic sampling methods include under-sampling and over-sampling are adopted to eliminate or minimize rarity by altering the distribution of relational examples. Experiments show, with the help of attribute filter method, R-NB can get better results than those without that. And, experiments show that multi-relational classifiers with under-sampling methods can provide more accurate results than that with over-sampling methods considering the ROC curve.

Keywords: imbalanced dataset, multi-relational data mining (MRDM), sampling, mutual information, naive Bayes.

1 Introduction

The class imbalance problem is of crucial importance since it is encountered by a large number of domains of great environmental, and is shown, in certain cases, to cause a significant bottleneck in the performance attainable by standard classification methods which assume a balanced class distribution. There have been a number of attempts at dealing with the class imbalance problem.

Timeweaver [1] uses a genetic algorithm to predict very rare events while Carvalho and Freitas [2; 3] use a genetic algorithm to discover "small disjunct rules." Raskutti uses support vector machines and learns only from the positive (rare) examples, thus recognizing patterns amongst the positive examples, rather than differentiating between positive and negative examples [4]. Typically the class distribution is altered to reduce the problems associated with rare classes, but the distribution of cases can also be altered to deal with rare cases [5; 6].Evaluation metrics that take rarity into account can improve data mining by better guiding the search process and better evaluating the end-result of data mining. Perhaps the most common is ROC analysis and the associated use of the area under the ROC curve (AUC) to assess overall classification performance [7; 8].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 562–570, 2008.
© Springer-Verlag Berlin Heidelberg 2008

These attempts are based on the assumption that all data resides, or can be made to reside, in a single table (relation). In recent years, new data analysis paradigm, called Multi-Relational Data Mining (MRDM) is proposed, which is the multi-disciplinary field dealing with knowledge discovery from relational databases consisting of multiple tables. Then, how to classify directly in multi-relational data sets with class imbalance?

Multi-relational naïve Bayesian classifier is one of the simplest methods which integrate statistical learning with relational learning to deal with classification task. To solve the class imbalance problem, this paper provides a multi-relational Naive Bayesian classifier named R-NB. To achieve higher performance, the attribute filter criterion based on mutual information is upgraded to deal with multi-relational data directly and the basic sampling methods include under-sampling and over-sampling are adopted to eliminate or minimize rarity by altering the distribution of relational examples. Experiments show R-NB and Graph-NB [9], with the help of attribute filter, can get better ROC curve than those without attribute filter. And, the classifier with under-sampling methods can provide more accurate results than that with over-sampling methods considering the ROC curve.

The remainder of the paper is organized as follows: Section 2 describes the R-NB method, upgrading of mutual information and the sampling method. Section 3 discusses the methodology used in the experiments, as well as the results achieved. Finally, Section 4 presents the conclusions and outlines future research.

2 Multi-Relational Classification in Imbalanced Domains

To solve the class imbalance problem, this paper proposes a multi-relational Naive Bayesian classifier named R-NB.

2.1 R-NB

Naive Bayes is a statistical classification method. It is simple but usually performs well even compared with many complicated methods. Originally it only deals with the data in one single table. To make it capable of dealing with multiple tables, there two types of methods. One is converting multiple tables into one table, and the other is dealing with each table separately. For the former method, a simple method is to join all tables together. In this paper, we do the latter.

Suppose T is the target table (the table containing the class label attribute), and R is another table that can be joined with table T. Table T has m attributes and table R has n attributes. For a tuple x in table T: $x = (x_1, x_2, \cdots x_m)$, there are l tuples in table R which can be joined with x. These l tuples are $(y_1, y_2, \cdots, y_l)$, where each tuple y_i is presented by n values: $y_i = (y_{i1}, y_{i2}, \cdots, y_{in})$.

Naive Bayes gives the assumption that the values of different attributes are independent given the class label. We adopt the extending assumption that given the class label, each tuple $(y_1, y_2, \cdots, y_l)$ from table R that can join with T is independent. Then, the multiple relational Naive Bayesian formula of R-NB is:

$$C_{MAP} = \arg\max_{c_i \in C} p(x, y_1, y_2, \cdots, y_l \mid c_i) p(c_i)$$
$$= \arg\max_{c_i \in C} \prod_{j=1}^{m} p(x_j \mid c_i) \prod_{q=1}^{l} \prod_{o=1}^{n} \frac{p(y_{qo} \mid c_i)}{l} p(c_i) \quad (1)$$

We estimate the probability in Formula 1 according to frequency of attribute occurrences. To avoid occurrence of a zero probability, we adopt Laplace correction to estimate probability.

2.2 Upgrading of Mutual Information

To achieve higher performance, the attribute filter criterion based on mutual information, which is synchronously considering two-class examples, is upgraded to deal with multi-relational data directly.

For discrete/categorical variables, the mutual information *I* of two variables *x* and *y* is defined based on their joint probabilistic distribution *P(x, y)* and the respective marginal probabilities *P(x)* and *P(y)*: (only discrete variables is considered in this paper)

$$I(x, y) = \sum_{i,j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i) p(y_j)} \quad (2)$$

For a multiple relational learning task, can the attributes from different tables be equally deal with? If we assign different weight for attributes form different table, how to decide weights? In fact, it is difficult for us to assign a weight for each table, because we usually know little about the domain. For simplicity, we assume that attributes from different tables have the same effect on classification. With this assumption, our attribute selection steps based on mutual information as follows:

Step1: compute mutual information $I(x_i, c)$ between each attribute x_i from each relation and target class *c*.

Step2: order attributes in terms of $I(x_i, c)$, then select the top *m* (generally defined by user) attributes in the descent ordering for classification.

This paper implements MIR-NB (which is R-NB with the upgrading attribute filter criterion based on mutual information) and MIGraph-NB (which is Graph-NB with the upgrading attribute filter criterion based on mutual information).

2.3 Sampling

To decrease the overall level of class imbalance, this paper adopts two sampling methods including under-sampling and over-sampling. Under-sampling eliminates majority-class examples while over-sampling, in its simplest form, duplicates minority-class examples. Both of these sampling techniques make the rare class less rare.

In this paper, the majority class is under-sampled by randomly removing samples from the majority class population until the minority class becomes some specified percentage of the majority class. This forces the learner to experience varying degrees

of under-sampling and at higher degrees of under-sampling the minority class has a larger presence in the training set. In our experiments, if the minority class had 50 samples and the majority class had 500 samples and we under-sample majority, the majority class would end up having 300 samples. We over-sample the minority class by duplicating minority-class examples. In our experiments, if the minority class had 50 samples and the majority class had 500 samples and we over-sample minority, in the end, the minority class would include 120 samples.

3 Experimental Study

We performed experiments on two real world databases. For each continuous attributes, we discretize it into 10 intervals. In addition, accuracy is the result of ten-fold cross-validation. Accuracy places more weight on the common classes than on rare classes, thus we adopt the ROC curve to assess overall classification performance.

3.1 Mutagenesis Database

The first experiments database is Mutagenesis database which is a frequently used MRDM benchmark. It contains 4 tables, 124 positive examples, 64 negative examples and totals to15218 tuples. We use BK2 of this dataset [9]. Figure 1 shows the ROC curves of R-NB and Graph-NB.

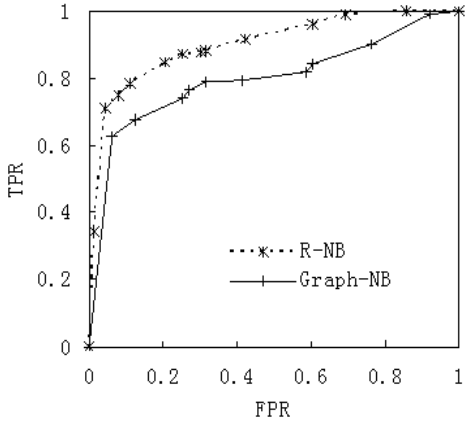


Fig. 1. ROC curves of R-NB and Graph-NB

Our experiments show that MIR-NB and MIGraph-NB have the same result when *m* is 3. Figure 2 shows the ROC curves of MIR-NB and MIGraph-NB when *m* is 3.

From Figure 1, it is easy to see that our new method R-NB is better than Graph-NB. Figure 2 indicates that the upgrading of mutual information can improve performance of classification algorithms.

For this database, we just give the results without sampling, since there are very little examples and the results have been good.

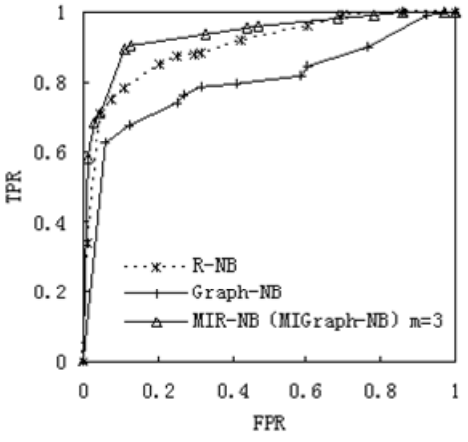


Fig. 2. ROC curves of MIR-NB and MIGraph-NB when m is 3

3.2 Financial Database

The second experiments database is the Financial database used in PKDD CUP1999. We do the simple modification to Table *trans*. Finally, this database has 7 tables, 606 positive examples, 76 negative examples and 75982 tuples totally.

Figure 3 gives the ROC curves of R-NB and MIR-NB (when m is 9) on Finance database without sampling. Figure 4 gives the ROC curves of Graph-NB and MI-Graph-NB (when m is 4) on Finance database without sampling.

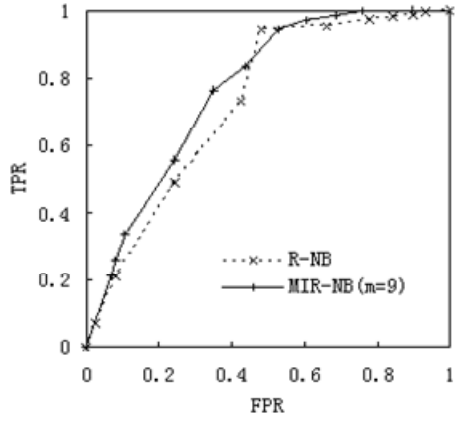


Fig. 3. ROC curves of R-NB and MIR-NB (when m is 9) on Finance database without sampling

From Figure 3 to 4, we can see multi-relational classification algorithms with up-grade of mutual information can get better results than those without that.

To decrease the overall level of class imbalance, we respectively adopt under-sampling and over-sampling on Financial database. The under-sampling method makes the database include 324 positive examples and 64 negative examples. The over-sampling method makes the database include 606 positive examples and 152 negative examples.

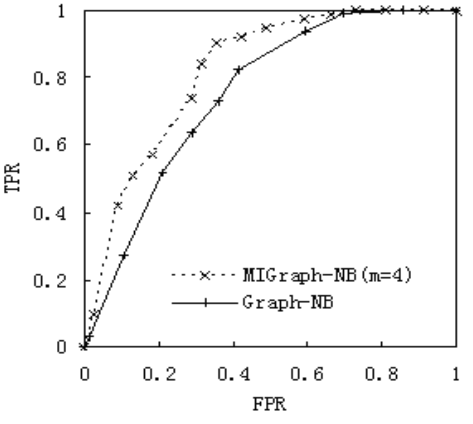


Fig. 4. ROC curves of Graph-NB and MIGraph-NB (when m is 4) on Finance database without sampling

Figure 5 gives the ROC curves of Graph-NB considering of three deferent conditions. Graph-NB1 shows the ROC curve of the database without sampling. Graph-NB2 shows the ROC curve of the database with over-sampling. Graph-NB3 shows the ROC curve of the database with under-sampling.

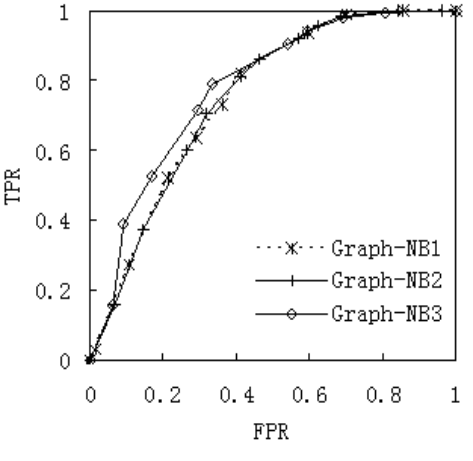


Fig. 5. ROC curves of Graph-NB considering of three deferent conditions

Figure 6 gives the ROC curves of R-NB considering of three deferent conditions. R-NB1 show the ROC curves of the database without sampling. R-NB2 show the ROC curves of the database with over-sampling. R-NB3 show the ROC curves of the database with under-sampling.

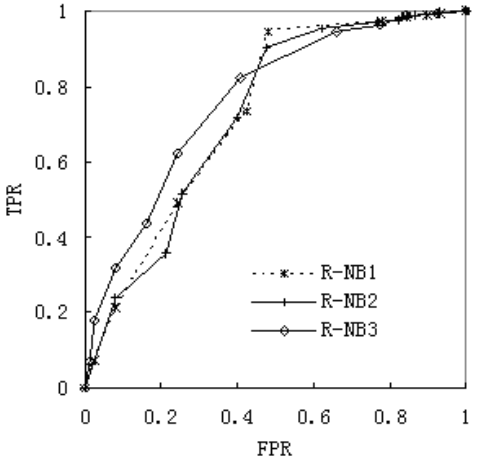


Fig. 6. ROC curves of R-NB considering of three deferent conditions

From Figure 6 to 7, we can see under-sampling can get the best results. We respectively give the results of the two algorithms of sampling combined with attribute filter in Figure 7 and 8. MIGraph-NB1 shows the ROC curve of the database without sampling (when m is 4). MIGraph-NB3 shows the ROC curve of the database with under-sampling (when m is 6). MIR-NB1 shows the ROC curve of the database

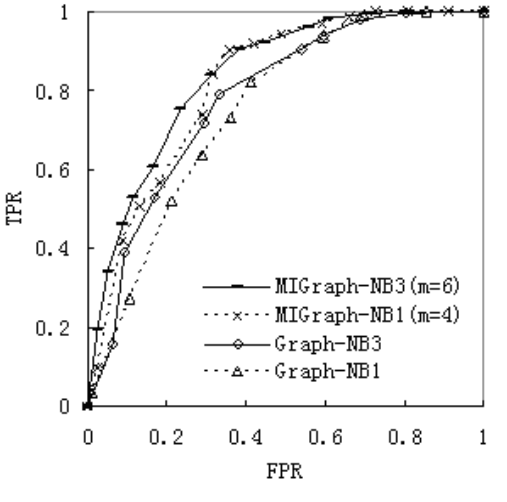


Fig. 7. ROC curves of Graph-NB considering of four deferent conditions

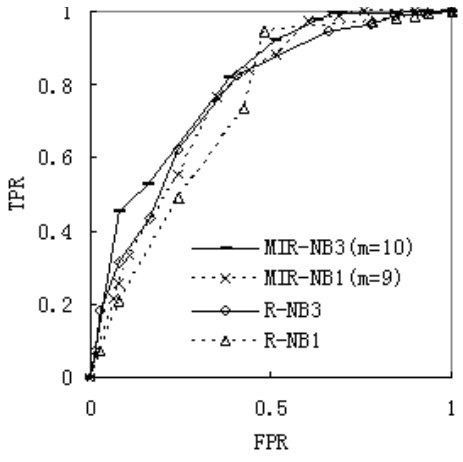


Fig. 8. ROC curves of R-NB considering of four deferent conditions

without sampling (when m is 9). MIR-NB3 shows the ROC curve of the database with under-sampling (when m is 10).

From Figure 7 and 8, we see that it can effectively improve performance to under-sample the database combined with the attribute filter method.

4 Conclusions

This paper discusses the problem of multi-relational classification on imbalanced datasets. To solve the class imbalance problem, this paper provides a new multi-relational Naive Bayesian classifier named R-NB. To achieve higher performance, the attribute filter criterion based on mutual information is upgraded to deal with multi-relational data directly and the basic sampling methods include under-sampling and over-sampling are adopted to eliminate or minimize rarity by altering the distribution of relational examples.

Experiments on Mutagenesis database show R-NB can give better ROC curves than Graph-NB. Experiments on two databases show R-NB and Graph-NB, with the help of attribute filter, can get better results than those without attribute filter. And, experiments on Financial database show that multi-relational classifiers (R-NB and Graph-NB) with under-sampling methods can provide more accurate results than that with over-sampling methods considering the ROC curve.

As future work, we plan to design more advanced sampling method and more appropriate evaluation metrics on multi-relational classification domains.

Acknowledgments

This work was supported in part by the Beijing Computer Science and Technology emphasis construction subject.

References

1. Weiss, G.M.: Timeweaver: a genetic algorithm for identifying predictive patterns in sequences of events. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 718–725. Morgan Kaufmann, San Francisco (1999)
2. Carvalho, D.R., Freitas, A.A.: A genetic algorithm for discovering small-disjunct rules in data mining. Applied Soft Computing 2(2), 75–88 (2002)
3. Carvalho, D.R, Freitas, A.A.: New results for a hybrid decision tree/genetic algorithm for data mining. In: Proceedings of the Fourth International Conference on Recent Advances in Soft Computing, pp. 260–265 (2002)
4. Raskutti, B., Kowalczyk, A.: Extreme re-balancing for SVMs: a case study. In: Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning (2003)
5. Japkowicz, N.: Concept learning in the presence of betweenclass and within-class imbalances. In: Proceedings of the Fourteenth Conference of the Canadian Society for Computational Studies of Intelligence, pp. 67–77. Springer, Heidelberg (2001)
6. Weiss, G.M.: Learning with rare cases and small disjuncts. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 558–565. Morgan Kaufmann, San Francisco (1995)
7. Bradley: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145–1159 (1997)
8. Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning 42, 203–231 (2001)
9. Liu, H., Yin, X., Han, J.: An efficient multi-relational naive bayesian classifier based on semantic relationship graphs. In: Proceedings of the 4th international workshop on Multi-relational mining, Chicago, Illinois, pp. 39–48 (2005)

Sandstorm Occurrence Frequency Short-Term Prediction Based on Bootstrap Method

Jianli Dong[1], Haiyang Chen[2], Jianzhou Wang[2], and Donghuai Sun[3]

[1] School of Computer Engineering, HuaiHai Institute of Technology, Lianyungang, 222005, China
dongjl1019@sina.com
[2] School of Mathematics & Statistics, Lanzhou University, Lanzhou, 730000, China
[3] Key Laboratory of Western China's Environmental Systems (Ministry of Education) College of Earth and Environment Sciences, Lanzhou University, Lanzhou, 730000, China

Abstract. Sandstorm is a disastrous weather phenomenon, Gansu Province is in the high incidence region of sandstorm. So the sandstorm is one of the major casualty weather in recent years in Gansu, causing lots of losses. it is undoubtedly favor of reducing loss to predict annual occurrence frequency of sandstorm. This paper based on achievements on sandstorm, choosed Dingxin, Jinta, Jiuquan, Zhangye, Minqin, five areas, where the sandstorm disasters is very serious, selected the latest 8 years' data by analysis the data of 1971 – 2007 as the original data, predict annual sandstorm occurrence frequency of recent years based on bootstrap method.

Keywords: Sandstorm, Short-term Prediction, Occurrence Frequendy, Bootstrap.

1 Sand Storm and Research Present Situation

1.1 The Sand Storm and the Influence to Humanity

The sand storm is the general name called for both the sand storm and dust storm, is a kind of calamitous weather phenomenon ,which means that strong wind draws much sand and dust near the ground surface into the sky, making air very foul and with visibility lower than 1km [9]. It affects people's normal life and activity. It occurred frequently is important indicator to the deteriorated environment, and speed up the land desertification greatly. It has caused lots serious harm to industry and agriculture in China. Because it can transport the lots of soil to the far place, the strong sand storm event is causes regional, even the global environment question and the climatic change question, brought critical impact to public health, even to the social economy development.

1.2 Present Research Situation for Sand Storm

With the development of the society and economy, people's awareness for environment protection environmental has been increased. The sand storm disaster has become the new focus which takes more and more public attention seriously. In recent years the

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 571–580, 2008.
© Springer-Verlag Berlin Heidelberg 2008

research about sand storm increased gradually and has made some new progress and achievement [4][1].

Wang Shigong [5] by systematic analysis the macroscopic synoptic climatology conditions and underlying surface situation of sand storm, conclude that there are five 5 aspect reasons for the sand storm mainly occurs in the spring and the early summer: 1 abundant sand dust source underlying surface and special topographical condition; 2 after long time drying and freezing in the winter, and defrosted in the spring , the surface soil became very loosen; 3 the position of spring upper-tropospheric jet axis is the substantial clause that the northern of China easy to blow the gale; 4 in spring, the atmospheric stratification instability increases, afternoon easy to generate convection to help the upper momentum download; 5 The spring is the northern area cold front moves the most frequent season.

The activity of cold font is the most frequent in spring at northern China, the gale after cold font is one of the important clauses for sand storm.

To reduce the harm of sand storm, we must control desertification firstly. While to removal lands from cultivation for reforestation, to reduce number of animals sustained by the pasture, and to take the appropriate artificial measures, restores the natural prairie vegetation, can be curbed the main source of sand and dust effectively and form green barriers. it require long time to restores the vegetation in big area, therefore, we must establish and improve the dynamic monitor and early warning systems of sand and dust weather, complete scientific research on disaster prevention and reduction to reduce dust weather-caused losses.

At present, in view of Chinese sand dust weather characteristic, we are mainly using detect and monitor ways, such as the meteorological satellite, the altitude detection, earth surface automatic weather station etc. to realize track and quantitative monitoring on occurrence, development and transmission of sand storm. The sandstorm forecast is based on traditional weather analysis and the personal understanding on numerical prediction products with the experience of predictors.

About the study on sand storm occurrence frequency, Zhang Li [7] analyzed the daily, seasonal, annual frequency evolution regularity in north China from 1954 to 2001 based on the ground meteorological observation data, as well as the correlativity between the regularity and wind velocity, relative humidity, precipitation, temperature, aridity. The result indicated that the sand storm occurrence days per year has the wave declining trend, especially in spring. In the about last 50 years, the direct natural causes of sandstorm frequency significant decline trend in northern China are the reduction of the wind velocity and the gale day number in sandstorm source area and occurrence area, and the precipitation increases of the main sand dust source area especially in spring and last winter, as well as, the improvement of atmospheric and the soil moist degree by the precipitation increases.

Wang Jing, in her master thesis [17] discussed the relationship between climatic factors and the sandstorm occurrence frequency, pointed out that the sand storm occurrences has significant correlation with the gale, and has rough inverse correlation with the antecedent precipitation.

This paper will predict sand storm occurrence frequency in the future use with a new statistical method named Bootstrap.

2 Introduce to Bootstrap Method and Bootstrap Interval Estimation

2.1 Introduce to Bootstrap Method

Bootstrap method, first introduced by US Stanford University statistics is Professor Efron.B [11], is a kind application-oriented statistical inference method which take the original data as the foundation, based on the large number of computing to simulate sampling, applied to study some statistics distribution characteristic of the original data, be suitable for those issues such as interval estimation, hypothesis test and so on, especially for those hard to solve with conventional method. It is considered one of new breakthroughs in statistics. Its essence is to simulate the process to draw a large number of samples from the original data randomly. It is essential to use Bootstrap that the original sample must reflect real distribution well.

Let the original samples $X = (X_1, X_2, \cdots, X_n)$ be independent identical distribution (i.i.d) samples, and $t(X)$ is an unbiased efficient estimation of some statistic. Firstly, get bootstrap samples by re-sampling with replacement from the original samples, in the resample, the probability for each sample point occurrence is equal, that is $P(\xi = X_i) = \frac{1}{n}$. A sample point of original samples is possibly to appear many times in one Bootstrap sample, also possibly never to appear. Repeat this process B times, then got B bootstrap samples $X_B^k, k = 1,2,\cdots,B$. We can get an estimation value with per bootstrap sample. So the bootstrap estimation of $t(X)$ is:

$$t_B(X) = \frac{1}{B}\sum_{k=1}^{B} t(X_B^k)$$

.

Efron. B [13] pointed out: in practical application, just let $B \geq 1000$, you can obtain the quite precise estimation or the reasonable inference result.

2.2 Interval Estimation

There are lots of studies on applying Bootstrap method in interval estimation. Efron.B [12] mentions several commonly used forms, Tibshirani [14] discussed these methods in detail. Beran.R[10] discussed their consistency. This paper uses Percentile Method. When we estimate confidence interval of statistic $t(X)$ in bootstrap methods, we also need draw B bootstrap samples from the original samples. Then we got the estimation value set $t(X_B^k), (k = 1,2,\cdots,B)$, which reflects distribution characteristics of $t(X)$. Structure the bootstrap empirical distribution of $t(X)$ with bootstrap samples, $t(X_B^k), (k = 1,2,\cdots,B)$.

$$\hat{F}(x) = P(t(X) \le x) = \frac{\#(t(X_B^k)_{k=1}^B}{B}$$

We give the confidence interval of $t(X)$ is $(\hat{t}_{\frac{\alpha}{2}}, \hat{t}_{1-\frac{\alpha}{2}})$, whose confidence level is $1-\alpha$, where $\hat{t}_{\frac{\alpha}{2}}$ is $\frac{\alpha}{2}$ - percentile, $\hat{t}_{1-\frac{\alpha}{2}}$ is $1-\frac{\alpha}{2}$ - percentile . Actually, we sort the bootstrap samples $t(X_B^k), (k = 1,2,\cdots,B)$ of $t(X)$, if the result is

$$t(X_B^{(1)}) \le t(X_B^{(2)}) \le \cdots \le t(X_B^{(B)}) .$$

It is obvious that $\frac{\alpha}{2}$ - percentile is $t(X_B^{([B*\frac{\alpha}{2}])})$, $1-\frac{\alpha}{2}$ - percentile is $t(X_B^{([B*(1-\frac{\alpha}{2})])})$. To improve the accuracy further, repeat this process R times, then got R confidence interval of $t(X)$ denoted by $(\hat{t}^s_{\frac{\alpha}{2}}, \hat{t}^s_{1-\frac{\alpha}{2}})$, $(s = 1,2,\cdots,R)$. Finally the confidence interval of $t(X)$ with level $1-\alpha$ is

$$(\frac{1}{R}\sum_{s=1}^{R} t^s_{\frac{\alpha}{2}}, \frac{1}{R}\sum_{s=1}^{R} t^s_{1-\frac{\alpha}{2}}) .$$

3 Sand-Storm Frequency Prediction

3.1 Sand-Storm Frequency Prediction

We select Dingxin, Jinta, Jiuquan, Zhangye, Minchin five regions in Gansu Province to study the sand storm occurrence in this paper, and attempt to make short-term prediction in these regions. The data is from the Gansu province meteorological observing station.

The occurrence frequency of the five regions show obvious wave declining trend. We can obtain the frequency mean value of the previous 29 year and the final 8 year not equal through hypothesis testing. This indicated that the sand storm occurrence frequency stage distinctness is significant. If we do not consider that this distinctness, predict the occurrence frequency with all data from1971 to 2007, it will certainly have errors.

Because the stage distinctness is obvious, we consider dividing the data into several parts. Wang Shigong in literature [5] [15] thought that this volatility takes 10 years as a period. But in view of the fact that Zhao Jin takes 5-7 years as predict term in literature [8], and the frequency disperses distribution figure, I think it is quite reasonable to take the period length not to surpass 10 years, that is 1971-1979, 1980-1989, 1990-1999, 2000-2020 are periods. The figures show that the points in a period gather, and the points in two different periods has big gap. So we consider that this partition is reasonable, and the sand storm occurrence frequencies are basically in the identical level in one period. In fact , this period division is very simple, the sand storm occurrence

frequency is stable in a stage, it will occur abrupt change in every year, and then will continues a new stable period based on this change point . There are few ways for abrupt change points detecting besides do it depend on the experience [3].

Considering that frequency period stability, we divided the period more reasonable based on the occurrence frequency figure (Figure 1. ~Figure 5.). This division is entirely artificial with aid of experience.

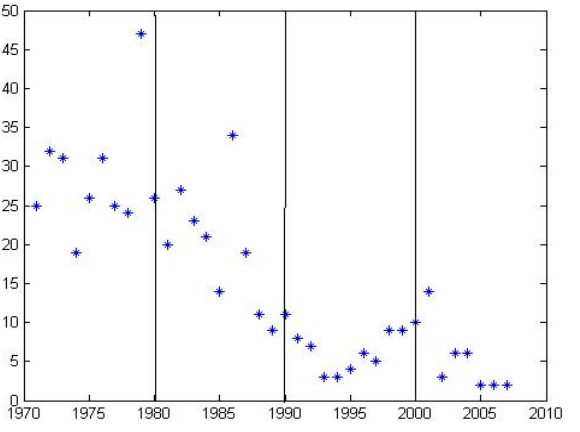


Fig. 1. Sandstorm Occurrence Frequency in Dingxin

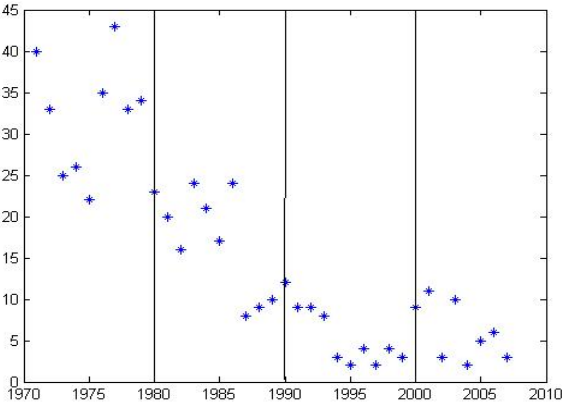


Fig. 2. Sandstorm Occurrence Frequency in Jinta

The result shows that:

1989-2007 is a stable period in Dingxin;
1987-2007 is a stable period in Jinta;
1987-2007 is a stable period in Jiuquan ;
1991-2007 is a stable period in Zhangye ;
1988-2007 is a stable period in Minqin .

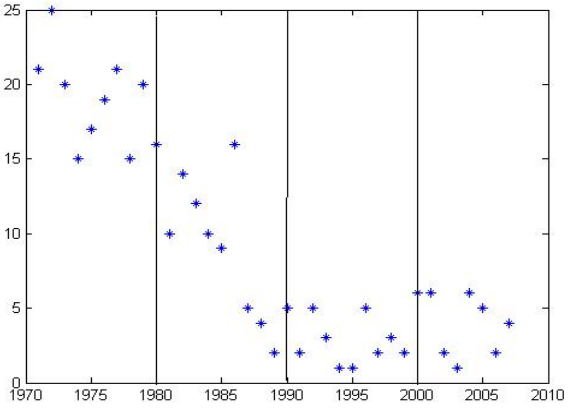


Fig. 3. Sandstorm Occurrence Frequency in Jinquan

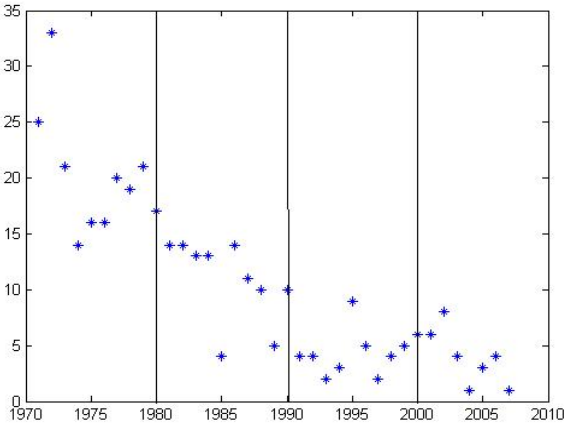


Fig. 4. Sandstorm Occurrence Frequency in Zhangye

From 2000 to 2007 are in a stable period, therefore we take the data from 2000 to 2007 (Table I) as the original data, that is we predict sand storm occurrence frequency for short-term based on the data of final 8 years.

3.2 Sandstorm Occurrence Frequency Short-Term Prediction

In a stable period, the sandstorm occurrence frequency always fluctuates around the mean value of this period frequency. Therefore we only need estimate the frequency mean value of this stable period, it can reflect frequency of next several year well. Now take the 1st district as an example to illustrate the steps. Let $\alpha = 0.05$, that is the confidence level is 95%.

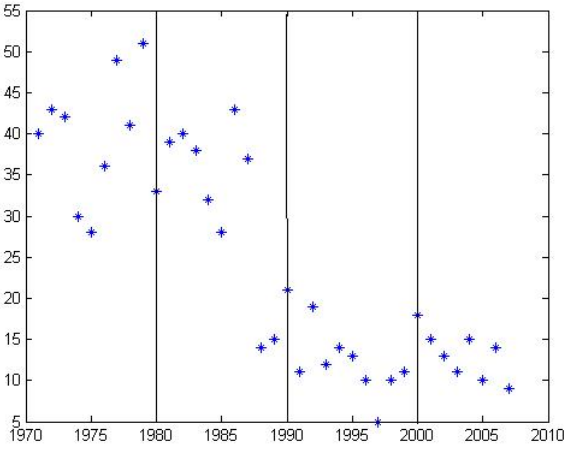


Fig. 5. Sandstorm Occurrence Frequency in Minqin

Table 1. Sandstorm occurrence frequency for final 8 years

	2000	2001	2002	2003	2004	2005	2006	2007
Dingxin	10	14	3	6	6	2	2	2
Jinta	9	11	3	10	2	5	6	3
Jiuquan	6	6	2	1	6	5	2	4
Zhangye	6	6	8	4	1	3	4	1
Minqin	18	15	13	11	15	10	14	9

1) Bootstrap re-sampling

Let $X = (X_1, X_2, \cdots, X_8) = (10,14,3,6,6,2,2,2)$ be the original sample, the data from 2000 to 2007, and the sample size $n = 8$. Resample with replacement B times from X, where $B = 1000$, each sample size is 8, which is consistent with the original sample size.

2) Obtain bootstrap empirical distribution

Let E be the mean of sandstorm occurrence frequency for this stable period. A estimation of E is

$$\hat{E}(X) = \frac{1}{n}\sum_{k=1}^{n} X_k .$$

Then we can get estimation value set of $\hat{E}_B$, that is $\{\hat{E}_B^s, s = 1,2,\cdots,1000\}$, based on all bootstrap samples $X_B^1, X_B^2, \cdots, X_B^{1000}$. So we obtain the bootstrap empirical

distribution of $\hat{E}_B$. Figure 6 show the distribution. It is obvious that this distribution is similar to normal distribution.

3) Confidence interval estimation

Sort all bootstrap estimation value of $\hat{E}_B$, we obtain the $\frac{\alpha}{2}$- percentile and $1-\frac{\alpha}{2}$- percentile, then we obtain the confidence interval is $[3.1250,8.6250]$ under level $1-\alpha$.

Repeat above process R times, let R=100 here. The final confidence interval is $[3.0500,8.6500]$.

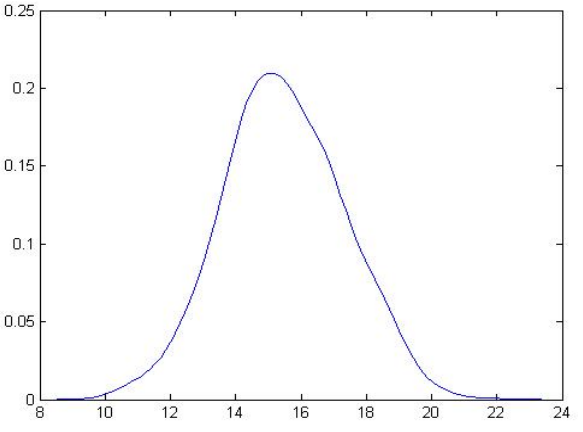


Fig. 6. Bootstrap empirical distribution of $\hat{E}_B$ B=1000

3.3 Prediction Result

Under the confidence level 95%, 97.5%, 99%, the sandstorm occurrence frequency estimation values and confidence intervals of Dingxin, Jinta, Jiuquan, Zhangye and Minqin sees in table 2.

Table 2. Prediction Results

	Mean	95%	97.5%	99%
Dingxin	5.0698	[3.05,8.65]	[2.75,9.07]	[2.50,9.66]
Jinta	6.0985	[3.92,8.42]	[3.66,8.68]	[3.33,9.04]
Jiuquan	3.9823	[2.63,5.28]	[2.46,5.43]	[2.20,5.61]
Zhangye	4.1095	[2.54,5.72]	[2.35,5.93]	[2.05,6.17]
Minqin	13.146	[11.20,15.07]	[10.95,15.32]	[10.62,15.62]

Because the sandstorm occurrence frequency must be integer number, we obtain the integer results through rounding . The results sees table 3.

Table 3. Prediction Results (Integer)

	Mean	95%	97.5%	99%
Dingxin	5	[3, 9]	[3, 9]	[3, 10]
Jinta	6	[4, 8]	[4, 9]	[3, 9]
Jiuquan	4	[3, 5]	[2, 5]	[2, 6]
Zhangye	4	[3, 6]	[2, 6]	[2, 6]
Minqin	13	[11,15]	[11,15]	[11,16]

The results show that the sandstorm occurrence frequency is probably $\hat{E}_B$, and the confidences inter under each level sees in table 3.

4 The Conclusion

1) Empirical distribution of sandstorm occurrence frequency is established in bootstrap method based on short-term historical simple data, has overcome the insufficient of small samples. And bootstrap method has strong objectivity.

2) To use Bootstrap method to estimate statistics only relies on with observation information assigned, does not need other suppositions and the additional observation, is ideal method carried on the statistical inference depend on the small sample .

3) It is a novel attempt to use bootstrap method to sand storm occurrence frequency prediction. The purpose is to explores a new method for sandstorm forcasting. In this area there are many issues to urgently await to be solved.

Acknowledgment

The authors would like to thank the referees for their most constructive comments. The research was supported by the NSF of China Grant 40625009, 90411012, and NSF of Gansu Province in China under Grant (ZS031-A25-010-G).

References

1. Li, Y.: New Advances of Research on Sand-dust Storm during Recent Years in China. Journal of Desert Research 24(5), 616–622 (2004)
2. Shi, X.: BOOTSTAP——A REVIVEW. Chinese Journal of Applied Probability and Statistics 3(2), 167–177 (1987)
3. Wang, J.: Several Problems Study of Climate Jump Detection and Climate Information Quality Supervising
4. Wang, S., Dong, G., et al.: Advances in Studying Sand-dust Storms of China. Journal of Desert Research 20(4), 349–356 (2000)
5. Wang, S., Dong, G., Yang, D., et al.: A Study on Sand-dust Storms Over the Desert Region in North China. Journal of Natural Disasters 5(2), 86–94 (1996)
6. Ye, D., Chou, J., Liu, J., et al.: Causes of Sand-stormy Weather in Northern China and Contral Measures. Acta Geographica Sinca 55(5), 513–521 (2000)

7. Zhang, L., Ren, G.: Change in Dust Storm Frequency And the Climatic Controls in Northern China. Acta Meteorologica Sinica 61(6), 744–750
8. Zhao, J., Xu, J.: Time-Series Fractal Characteristics of Sandstorm Frequency in Hexi Corrido. Journal of Desert Research 23(4), 415–419 (2003)
9. Zhao, X.: Damage and Countmeasures of Catastrophic Sandstorm Occurred in Gansu Province. Journal of Desert Research 13(3), 1–7 (1993)
10. Beran, R.: Bootstrap Methods in Statistics, Jber.d.Dt., Math-Verein, pp. 14–30 (1986)
11. Efron, B.: Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics 7(1), 1–26 (1979)
12. Efron, B.: The Jackknife, The Bootstrap and other resampling plans. SIAM, Philadelphia (1983)
13. Efron, B.: Jackknife-after-Bootstrap standard error sand influence functions. J.R. Statist., Soc.B 54, 83–127 (1992)
14. Tibshirani, R.J.: Bootstrap confidence intervals, Tech Report. No.91, Stanford University (April 1984)
15. Wang, S., Wang, J., Zhou, Z., et al.: Regional characteristics of three kinds of dust storm events in China. Atmospheric Environment 39, 509–520 (2005)
16. Wang, J.: Relation between the SandStorm Occurrence Frequeney and Climate Factors and the Analysis of Dust Entrainment and Deposition, Thesis Submitted to Nanjing University of Information Science & Technology inpartial fulfillment of the requirements for the degree of Master of Natural Science (2007)

Simplify Multi-valued Decision Trees

Chien-Liang Liu and Chia-Hoang Lee

Department of Computer Science
National Chiao Tung University
1001 Ta Hseuh Road, Hsinchu, Taiwan 300, ROC
clliu@mail.nctu.edu.tw

Abstract. Decision tree is one of the popular data mining algorithms and it has been applied on many classification application areas. In many applications, the number of attribute values may be over hundreds and that will be difficult to analyze the result. The purpose of this paper will focus on the construction of categorical decision trees. A binary splitting decision tree algorithm is proposed to simplify the classification outcomes. It adopts the complement operation to simplify the split of interior nodes and it is suitable to apply on the decision trees where the number of outcomes is numerous. In addition, meta-attribute could be applied on some applications where the number of outcomes is numerous and the meta-attribute is meaningful. The benefit of meta-attribute representation is that it could transfer the original attributes into higher level concepts and that could reduce the number of outcomes.

1 Introduction

Over the last few decades, data mining has become a very active and important research area nowadays. Many data mining algorithms are proposed and have proven their capabilities on intelligent systems. Decision tree is one of the popular machine learning algorithms and it has been applied on many classification application areas. In essence, decision tree is simple to understand and the inference result is easy for humans to interpret. Besides, decision tree is robust and it performs well with large data set.

In general, decision tree construction includes tree-building phase and tree-pruning phase. In tree-building phase, the training data is recursively partitioned until all the records in a partition have the same class. In each partition, a new attribute should be selected as the splitting criterion and new sub-partitions are collected according to the splitting criterion. The same process is applied on these sub-partitions recursively. Appropriate attributes have to be selected as the root nodes in each partition and there are several approaches to the selection of attributes. ID3 [1] adopts information entropy [2] and information gain on the selection of attributes. C4.5 [3], which is an extension of ID3 that accounts for unknown values, continuous attribute values, pruning, and so on, adopts gain ratio as the selection criteria. CART [4] (Classification and Regression Trees), SLIQ [5] and SPRINT [6] select the test with the lowest GINI index. In practice,

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 581–590, 2008.
© Springer-Verlag Berlin Heidelberg 2008

a perfect decision tree may lead to overfitting problem and that makes the error rate of testing data increase. Many pruning approaches are proposed in tree-pruning phase to prevent overfitting of the training data. The pruning approaches include MDL pruning ([7][8][9]), cost-complexity pruning ([10]), and pessimistic pruning ([10]). In addition to prune in the tree-pruning phase, Garofalakis et al. [11] proposed integrated decision tree construction algorithms that push size and accuracy constraints into the tree-building phase.

The purpose of this paper will focus on the categorical decision tree. A binary splitting decision tree algorithm is proposed to simplify the classification outcomes. It adopts the complement operation to simplify interior nodes splitting and it is suitable to apply on the decision trees where the number of outcomes is numerous. When the complement split is applied on the attribute value test, the number of outcomes will become two. The benefit of this approach is that it could provide a simplified model on multi-valued decision trees, since the number of outcome will decrease and the rules description will contain set element. Moreover, the complement split operation could be incorporated with existing decision tree algorithms. The complement split operation could be applied at specific nodes and the other nodes are applied original split operations. With the incorporation, it may help users discover new rules due to the set elements in the rules.

In addition, meta-attribute could be applied on some applications where the number of outcomes is numerous and the meta-attribute is meaningful. The benefit of meta-attribute representation is that it could transfer the original attribute into higher level concept and that could reduce the number of outcomes. In this paper, a meta-attribute algorithm is proposed to transform the categorical attribute values into digital numbers and the range of the digital numbers are used to indicate the meta-attributes.

2 Preliminaries

Traditionally, the induction biases used in decision tree is to employ Occam's razor. Smaller trees are preferred, because they are easier to understand and may also have higher predictive accuracy. In essence, different attribute selection sequences will lead to different decision trees. Hence, the attribute selection criteria plays an important role in tree-building phase. Although the goal is to find the smallest tree, it is a NP-complete problem to find the smallest one. In practice, greedy algorithms based on local search are used to yield reasonable trees in reasonable time. In the following section, two decision tree construction mechanisms are described.

2.1 ID3

The goal of classification decision trees is to construct a classifier from the given training data and predict the class label of unknown data. The classifier classifies the training data based on the homogeneous level of the data. The more similar they are, the most likely they will be grouped on the same class. Thus, the

attribute which could classify the data into homogeneous group will be preferred when constructing decision trees. ID3 adopts information gain to evaluate the level of homogeneous among the data. The one with the highest information gain is selected as the interior node. In theory, the information gain is based on entropy which could be used to measure the uncertainty of a random variable. The entropy $H(X)$ of a random variable X is defined by:

$$Entropy(X) = \sum_{x \in X} P(x) \log P(x)$$

In decision tree, supposed that there are N_m instances reaching node m, and N_m^i belongs to class C_i. The entropy of node m will be I_m: (p_m^i is the probability of instances belonging to class C_i)

$$p_m^i = \frac{N_m^i}{N_m}$$

$$I_m = -\sum_{i=1} p_m^i \log p_m^i \tag{1}$$

In ID3, the information gain is based on the impurity after the splitting. Assume that the number of instances taking branch j is N_{mj} and N_{mj}^i belong to class C_i. The impurity after the splitting will become $I_m^{'}$: (k means the number of class and l means the number of branch)

$$p_{mj}^i = \frac{N_{mj}^i}{N_{mj}}$$

$$I_m^{'} = -\sum_{j=1}^{l} \frac{N_{mj}}{N_m} \sum_{i=1}^{k} p_{mj}^i \log p_{mj}^i = \sum_{j=1}^{l} \frac{N_{mj}}{N_m} I_{mj} \tag{2}$$

The information gain of the split at node m is the difference between I_m and $I_m^{'}$, where I_m and $I_m^{'}$ are obtained from Equation 1 and Equation 2 respectively. Each attribute could obtain its information gain and the highest one will be selected as the interior node.

In each partition, ID3 follows information gain criterion recursively to choose the interior node until the partition belongs to the same class or there are no new instances left. The drawback of ID3 is that it favors the attribute with many outcomes when applying information gain. C4.5 [3] overcomes this problem by using normalization on information gain.

2.2 C4.5

C4.5 is an extension of ID3 that accounts for unknown values, continuous attribute values, pruning, and so on. It adopts gain ratio as the selection criteria to overcome some drawbacks of ID3. It takes into account the number of the elements in the subset and it defines a variable “Split Info”.

$$SplitInfo(m) = -\sum_{i=1}^{k} \frac{|N_m^i|}{|N_m|} \times \log_2(\frac{|N_m^i|}{|N_m|})$$

This represents the potential information generated by dividing N_m into k subsets, whereas the information gain measures the information relevant to classification that arises from the same division. Then, gain ratio expresses the proportion of information generated by the split that is useful and it is defined as follows:

$$GainRatio(m) = gain(m)/SplitInfo(m)$$

The attribute with the maximum gain ratio is selected as the interior node. According to Quinlan's experiment [3], the gain ratio criterion is robust and typically gives a consistently better choice of test than the gain criterion.

3 Multi-valued Decision Tree

3.1 Scenario 1

In practice, when gain ratio criterion is applied on attributes selection, the attribute with many outcomes will reduce its overall gain ratio even though the attribute consists of some values that could divide the original data into homogeneous subsets. For example, as shown in Fig. 1, the node S includes 300 positive instances and 150 negative instances and it could be partitioned into 10 subsets (S_1, S_2,..., S_{10}) based on the attribute $A's$ values A_1, A_2,..., A_{10}. The number of positive and negative instances for each subset is listed below. In this example, when computing the gain ratio of attribute A at node S, two possibilities exist:

1. If the gain ratio of attribute A at node S exceeds other attributes, attribute A's values will be selected as the splitting criterion and it will partition the original set into 10 subsets.
2. If the gain ratio of attribute A at node S does not exceed other attributes, attribute A will not be selected even though the subset S_{10} is an approximately homogeneous group.

In this paper, a binary splitting algorithm is proposed to simplify the splitting and that could reduce the number of outcomes.

3.2 Scenario 2

In addition to scenario 1, meta-attribute characteristic exists in many applications. For example, when the system performs knowledge discovery process on the user profile database to find out the user behavior patterns, address information may be included in the analysis. In practice, city is one of the address attributes and the number of cities may be over hundreds. Instead of adopting city attribute directly, meta-attribute concept could be applied to reduce the number of attribute values. The grouping of the attribute value could make

the attribute value become more general concept and that may provide more meaningful information as well. In this paper, an algorithm which adopts meta-attribute to solve multi-values issues is proposed.

4 Binary Categorical Splitting

In this paper, a binary categorical splitting heuristic approach is proposed to reduce the number of subsets when the application contains attributes that consist of many outcomes. The approach proposed in this paper splits the original set into two subsets. One of the subsets is based on one of the attribute values and the other is based on its complement. The gain ratio is adopted to represent the homogeneous level. For the node containing attribute with N values, there will have N pairs. The highest gain ratio among these pairs will be selected. For example, in the tree structure of the Fig. 1, there are 10 pairs (A_1, $NOT\ A_1$), (A_2, $NOT\ A_2$), ..., and (A_{10} , $NOT\ A_{10}$) and their corresponding subsets are (S_1, $S - S_1$), (S_2, $S - S_2$), ..., and (S_{10}, $S - S_{10}$) respectively. Fig. 2 and Fig. 3 show the result when the splits are applied on A_1 and A_{10} respectively. The gain ratios of all the partitions could be computed and the highest one will be selected

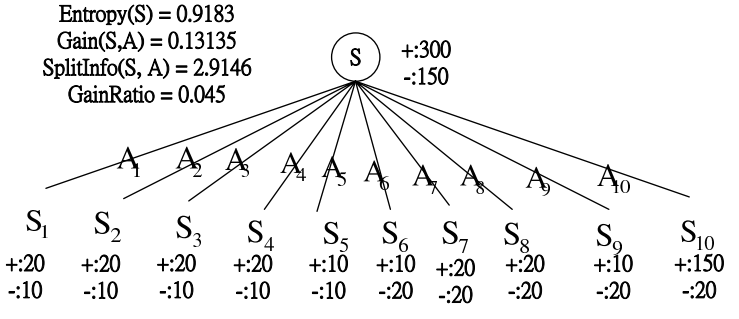


Fig. 1. Decision tree splitting node with 10 outcomes

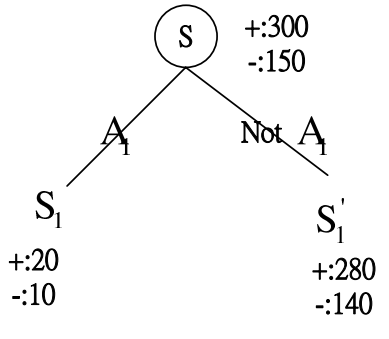


Fig. 2. Decision Tree Splitting on A_1 and Not A_1

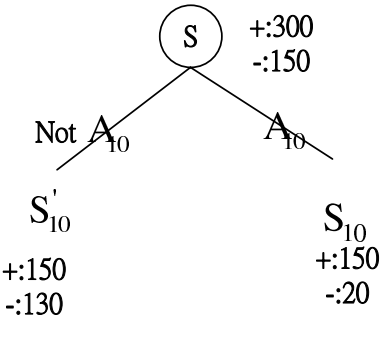


Fig. 3. Decision Tree Splitting on A_{10} and Not A_{10}

Algorithm 1. Decision Tree Generation Algorithm

procedure *GenerateTree*(X)

1. **if** NodeEntropy($X < \theta$) **then**
2. Create leaf labeled by majority class in X
3. **return**
4. **end if**
5. $i \leftarrow SplitAttribute(X)$
6. **for all** branches of X_i **do**
7. Find X_i falling in branch
8. $GenerateTree(X_i)$
9. **end for**

Algorithm 2. Decision Tree Attribute Split Algorithm

procedure *SplitAttribute*(X)

1. $MinEnt \leftarrow MAX$
2. **for all** attributes $i = 1, ..., d$ **do**
3. {assume that X_i is discrete with n values}
4. Split X into $X_1, ..., X_n$ by X_i
5. **for** $j = 1$ to n **do**
6. $e \leftarrow SplitEntropy(X_j, NotX_j)$
7. **if** $(e < MinEnt)$ **then**
8. $MinEnt \leftarrow e$
9. $bestf \leftarrow i$
10. **end if**
11. **end for**
12. **end for**
13. **return** $bestf$

as the splitting node. Therefore, A_{10} and $NOT\ A_{10}$ will be selected and the splitting becomes a binary categorical splitting and the gain ratio is higher than the original splitting. When this heuristic approach is applied on the nodes with many outcomes, it could simplify the number of outcomes and the complement operation sometimes may help users discover new concept group. Algorithm 1 and Algorithm 2 are the algorithms that construct the binary splitting decision tree. The advantages of this approach are as follows:

1. It could simplify the rules (including the number of rules and tree structure).
2. It may discover new concept groups from the complement of the attribute values.
3. Algorithm 2 could incorporate with other decision tree algorithms. Complement operation is only applied at specific nodes.

5 Meta-attribute Grouping

In addition to the binary categorical splitting approach described above, meta-attribute could be applied on the circumstances where the number of attribute

values is numerous and meta-attribute could provide meaningful attribute. For example, city information could be grouped into another higher level attributes such as province or area. In some countries, the number of cities is hundreds and that may lead to some biases during tree construction. If province is a higher concept of city, the city could be group into its corresponding province and the number of outcomes will decrease dramatically. Furthermore, province information could be grouped into another higher level concept such as area and the number of outcomes will be less than 10.

In addition, it could provide meta-knowledge information on the result due to the aggregation characteristic. However, these categorical data could not be grouped directly, so some preprocessing steps are required. The mapping steps are described as follows:

1. Make a mapping table to store the categorical names and their mapping digital numbers.
2. Order the attribute mapping digital numbers based on the grouping of their meta-attributes. As shown in Fig. 4, there are attribute values $a_1, ..., c_r$ and $a_1, a_2, ..., a_n$ belong to meta-attribute A, $b_1, b_2, ..., b_m$ belong to meta-attribute B and $c_1, c_2, ..., c_r$ belong to meta-attribute C. Thus, the mapping digital number of $a_1, a_2, ..., a_n$ could be defined as $1, 2, ..., n$ or the other group of consecutive digital numbers. The mapping digital number of $b_1, b_2, ..., b_m$ could be defined as $n + 1, n + 2, ..., n + m$ or the other continuous digital numbers that do not intersect with the other groups.
3. Transform these categorical attribute values in the data set into their mapping digital numbers and store the mapping information in the mapping table created in Step 1.

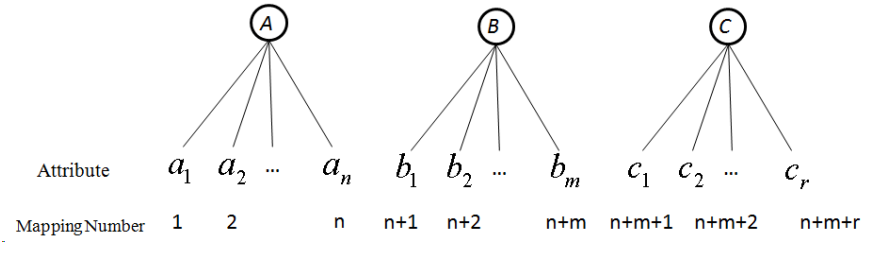


Fig. 4. Mapping between the attribute values and the digital numbers

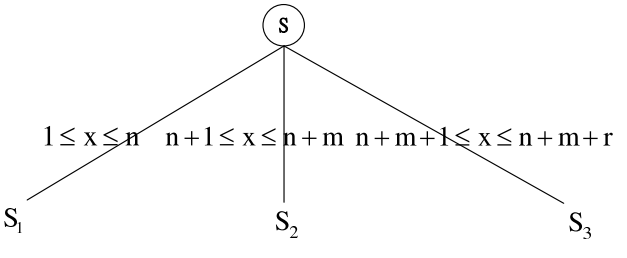


Fig. 5. Decision tree numerical attribute values

After the above mapping steps, the attribute within the same meta-attribute could be grouped together, so the range of the digital number could be used to represent splitting criterion. Therefore, every categorical attribute values could be dispatched to specific group by considering the range group of their corresponding digital number. As shown in Fig. 5, S_1, S_2 and S_3 represents meta-attribute A, B and C respectively.

6 Experiment

In this section, a "sunburned" example is used to demonstrate the use of complement attribute values. Table 6 shows that, there are name, hair, height, weight, lotion attributes and a class label (sunburned or none) for each record. Fig. 6 shows the result of decision tree using C4.5 algorithm and the number of rules is 4. In the other hand, the complement operation is applied and Fig. 7 shows the final result. Compare the final rules generated by these two models, two rules in Fig. 6 are replaced by one complement rule in Fig. 7. The original rules describe the color of hair in a specific way (red and blonde) while the complement one describes the color in a set way (not brown). Instead of specific meaning, set representation provides more general concept and that may help the users discover new concepts from the complement operation. Furthermore, the "SplitAttribute" procedure described in Algorithm 2 could be modified to incorporate with other decision tree algorithms. Binary categorical split is applied only when some criterion meets and the others adopt original splitting algorithms.

In many countries, the number of countries is over hundreds. In Taiwan, there are 15 cities and there will have 15 branches if city attribute is selected as a splitting criterion. In general, Taiwan cities could be categorized by their geographical location. Based on the geographical location, the attribute name could be transformed into corresponding digital number as shown in Table 2. The location of the city could be identified from the range of "ID" value. For example,

Table 1. Sunburned Example Data

ID	Name	Hair	Height	Weight	Lotion	Result
1	Sarah	Blonde	Average	Light	No	Sunburned
2	Dana	Blonde	Tall	Average	Yes	None
3	Alex	Brown	Short	Average	Yes	None
4	Annie	Blonde	Short	Average	No	Sunburned
5	Emily	Red	Average	Heavy	No	Sunburned
6	Pete	Brown	Tall	Heavy	No	None
7	John	Brown	Average	Heavy	No	None
8	Katie	Blonde	Short	Light	Yes	None

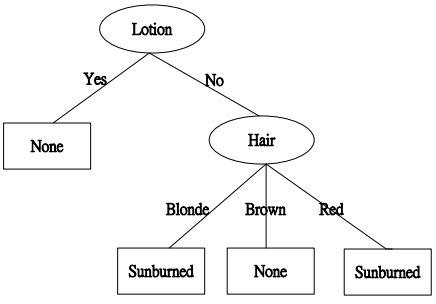


Fig. 6. Decision tree using C4.5

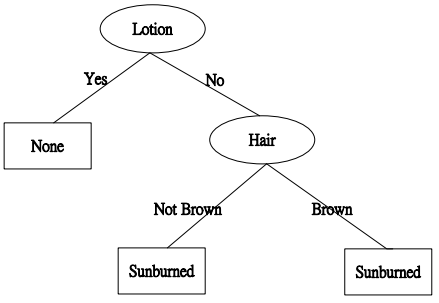


Fig. 7. Decision tree using complement operation

Table 2. Taiwan City List

City Name	ID	Location	City Name	ID	Location
Taipei	1	North	Yunlin	9	South
Yilan	2	North	Chiayi	10	South
Taoyuan	3	North	Tainan	11	South
HsinChu	4	North	Kaohsiung	12	South
Miaoli	5	North	PingTung	13	South
Taichung	6	Central	Taitung	14	East
ChangHua	7	Central	Hualien	15	East
Nantou	8	Central			

if the "ID" is less than 6, it could be transformed into "North". The number of branches of reduced decision tree becomes 4.

7 Conclusion

In this paper, two algorithms are proposed to reduce the number of outcomes when the number of attribute values is numerous. In the first binary categorical splitting algorithm, node splitting is based on attribute value and its complement value. Gain ratio is used to compute the homogeneous level and the highest pair will be selected as the splitting criterion. Instead of a specific attribute value splitting approach, it performs the splitting in a set way. In addition to the simplification of the tree structure, it may help users discover unknown concept from complement operations. Moreover, complement operation could also be incorporated with existing decision tree splitting algorithms to reduce the complexity of outcome at specific node.

As for meta-attribute grouping, some attributes could be grouped into higher level concept practically. In this paper, the categorical attributes are transformed into digital number and the order of these digital numbers are based on their meta-attribute concepts. Thus, the range of the digital number could be used to represent the meta-attribute. The benefit of this approach is that it could provide more conceptual representation and it could reduce the number of outcomes dramatically.

Acknowledgment

This work was supported in part by the National Science Council under the Grants NSC-97-2221-E-009-135 and NSC-97-2811-E-009-019.

References

[1] Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
[2] Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
[3] Quinlan, J.R. (ed.): C4.5: Programs for Machine Learning. Kaufmann Publishers Inc., San Francisco (1993)
[4] Breiman, L., Friedman, J., Stone, C.J., Olshen, R.: Classification And Regression Trees. Chapman & Hall/CRC, Boca Raton (1984)
[5] Mehta, M., Agrawal, R., Rissanen, J.: Sliq: A fast scalable classifier for data mining. In: Proc. of the Fifth International Conference on Extending Database Technology, pp. 18–32 (1996)
[6] Shafer, J.C., Agrawal, R., Mehta, M.: Sprint: A scalable parallel classifier for data mining. In: Proc. 22nd Int. Conf. Very Large Databases, VLDB, pp. 544–555 (1996)
[7] Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1029 (1993)
[8] Mehta, M., Rissanen, J., Agrawal, R.: Mdl-based decision tree pruning. In: Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD 1995), pp. 216–221 (1995)
[9] Quinlan, J.R., Rivest, R.L.: Inferring decision trees using the minimum description length principle. Information and Computation 80(3), 227–248 (1989)
[10] Quinlan, J.R.: Simplifying decision trees. International Journal of Man-Machine Studies 27(3), 221–234 (1987)
[11] Garofalakis, M., Hyun, D., Rastogi, R., Shim, K.: Building decision trees with constraints. Data Mining and Knowledge Discovery 7(2), 187–214 (2003)

A GIS-Based Evaluation on Sensitivity of Soil Erosion in Yishusi River Watershed*

Jia Yan, Jun Du, Chongsheng Xue, and Qinghua Yang

Faculty of Earth Sciences, China University of Geosciences,
430074 Wuhan, China
zslq01@cug.edu.cn, dujun_20002@yahoo.cn,
xuechsh@gmail.com, qinghua9@yahoo.cn

Abstract. Soil erosion sensitivity is limited by a number of factors, including precipitation, relief, soil quality, vegetation and human activity. With reference to domestic and international work on the assessment of sensitivity to soil erosion and to the natural environment characteristics of Yishusi River watershed, the primary influences on and sensitivity grades of soil erosion are determined. The Soil erodibility factor and vegetation factor values used for soil erosion sensitivity evaluation are obtained by interpreting Landsat TM image data, the rainfall erosion force factor map and slop map are obtained by Digitizing and space analyzes in GIS; We evaluated the factors controlling sensitivity to soil erosion and classified areas according to sensitivity by analyzing the reasons for erosion by GIS. The space distributing of all classificatory soil erosion sensitivity evaluation of Yishusi River watershed is analysized. The results showed that 15.74 percent of the total land(approximately 4987.88km2) are maintaining at high or extremity sensitivity, 53.12 percent of the total land(approximately 16836.39km2) are maintaining at moderate sensitivity,31.14 percent of the total land(approximately 9870.88km2) are maintaining at mild or none sensitivity; it concludes that entire soil erosion sensitivity of Yishusi River watershed is high. At last, this article gives some suggestions and measures for the worse situation. This article is able to support the practice work of soil erosion prevention, monitoring management as well as the land use in watershed. Meanwhile, the development of the index system can also benefit the theory of watershed planning and management.

Keywords: Yishusi River watershed, soil erosion, sensitivity, GIS.

1 Introduction

Eco-environmental sensitivity refers that sensitive degree of partial eco- environmental elements or factors are responded through effecting by human activities, generally we make a quantitative evaluation for arisen probability degree of regional eco-environmental problems [1]. OUYang,Z.Y and Wang,X.K pointed out that eco-environmental sensitivity refer to responsive degree of eco-environment through

* This work was supported by the Zhongkai Mining Foundation.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 591–598, 2008.
© Springer-Verlag Berlin Heidelberg 2008

effecting by human activity and natural environment changes; it can show a degree of difficulty and possibility for arising regional eco-environmental problems. With increase and enhancement of regional and global eco-environmental problems day by day, eco-environmental sensitivity has been applied gradually in domestic and overseas. At present, most researchers concentrate their attention on qualitative research, lack quantitative research.

The study selected Yishusi River watershed area for the scope of the study, namely Yimeng Mountain Area which lie in the upper reaches of Yishusi River and east of Nansi lake, including Yi River, Shu River ,Si River and Futong River watershed area of eastern coast, the district include Linyi city,Zaozhuang city, parts of Zibo city and Rizhao city of Shandong province, the total areas of 1695.15km2, accounting for the total area of Shandong procince of 20.18%, accounting for the Huaihe River basin area of 11.74%. We used the combinative methods of qualitative and quantitative evaluation to evaluate soil erosion sensitivity. The research area was classified according to sensitivity which will provide scientific bases for the development of the corresponding control and economical restoration strategies.

2 Choice of Evaluation Factors and Classification Standards

There are many factors to effect soil erosion, different elements of environment under the action of same disturbance factors are not the same sensitive performance. At the same time same elements of environment under the action of different disturbance factors are not the same sensitive performance. Hereby, we construct the cross-impact matrix which set disturbance factors as row and element of environment as column; we give 5 grade of sensitivity to measure any known sensitivity of environmental element.

A great number of analysis and research about formation and effect of factors for soil erosion problems show that the sensitivity of the regional soil erosion is mainly affected by climate, soil characteristics, landform and surface coverage. So it can be used terrain slope, erosion of rainfall, soil erodibility, and surface vegetation coverage to evaluate [2].

(1) Rainfall erosion force: this index is an exogenetic force factors to reflect the precipitation caused soil erosion, it is also the important parameter to evaluate precipitation caused the latent capacity of soil erosion[4]. Yishusi River watershed area of Shandong province is located in china's monsoon climate of warm temperate zone, there is abundant precipitation, where average annual precipitation is about 830mm.The precipitation has obvious differences in the spaces and time and it is easy to product erosive effect to soil.

(2) Soil erodibility factor K: this index is an endogenic force, which is used to evaluate the difficult degree what soil is separated, eroded and conveyed, and it reflect the soil texture effect on soil erosion[5].

(3) Slope factor: Geomorphologic shape is the important element to effect soil erosion. The slope is primary element for soil erosion in the factors of landform. Studies show that soil erosion is positive correlation with slope size. And the steeper a slope, shorter a time of precipitation, the higher rate of soil erosion, and the greater amount of soil erosion.

(4)Vegetation coverage factor: Soil erosion occurred on small chance in the higher surface vegetation coverage region. On contrary, soil erosion occurred on big chance in the lower surface vegetation coverage area. Therefore, the surface vegetation coverage is the important factor to evaluate the sensitivity of soil erosion.

3 Research Method

In order to ensure that the space has good superposition of different thematic map layers, we must unity the coordinate system and projection system of evaluation factors map supported by GIS. We adopt the projection system of equivalent area dual-latitude secant conic projection, the unified central meridian is 105ºE, double standard latitude are 25ºN and 47ºN, use the ellipsoid of KRASOVSKY, the start point is equatorial 0º[3]. In addition, we want to use structure of raster date to achieve calculation of various algebra and logic under taking into account the ability of GIS spatial analysis, the vector format will be converted to GRID format by using POLYGRID, SPLINE and other commands of ArcGIS. The size of grid cell is 30m × 30m.

3.1 Establishment Graphics and Attribute Database of Single-Factor Sensitivity Evaluation

We established the graphics and attribute database of Rainfall erosion force, DEM, soil erodibility and surface vegetation coverage by using ArcEdit module of ArcGIS, and made evaluation of single-factor sensitivity based on graded standard（Fig.1）.

In order to objectively evaluate the impact of various factors, the factors would be selected for classification and quantification by using arithmetic function of ArcGIS, and convert vector data to raster data. The factor plays the important role in ecological and environmental sensitivity if its data is greater.

(1) Rainfall erosion force: First of all, we calculated and draw distribution maps of Rainfall erosion force and got the raster maps of Rainfall erosion force of this watershed area by using method of SPLINE interpolation. According to the graded standard of Rainfall erosion force, we get the map which shows Rainfall erosion force effect on soil erosion.

(2) Soil erodibility factor K: In this study we used method of checking chart to calculate the soil erodibility. First, we used the method of digitization to get the vectorial map of soil erodibility, and then we coveraget this vectorial map to GRID by using command of POLYGRID. According to graded standard, we assign the value for the map, finally we get the map which shows soil erodibility effect on soil erosion.

(3) Slope factor: Sploe information was extracted from 1:250000 DEM by using method of SPLINE of ArcGIS. According to graded standard, to assign the value for map and get the raster map which shows slope effect on soil erosion.

(4) Surface vegetation coverage factor: Using the remote sensing images of research area in 2000 can be interpreted to get the map of land use types. According to graded

standard of surface vegetation coverage, we can get the map which shows surface vegetation coverage effect on soil erosion.

3.2 Synthetical Evaluation of Sensitivity

Soil erosion sensitivity is impacted by aforesaid four factors' comprehensive action, so we make synthetical evaluation based on single-factor sensitivity evaluation. We applied the synthetical index to evaluate the current situation of soil erosion sensitivity for research region, using GRID module of ArcGIS to implement spatial overlay of map and calculative function, and then using Arcview software to draw the distribution map of soil erosion sensitivity, finally we would analyze the result.

$$SS_j = \sqrt[4]{\prod_{i=1}^{4} P_{ij}} \tag{1}$$

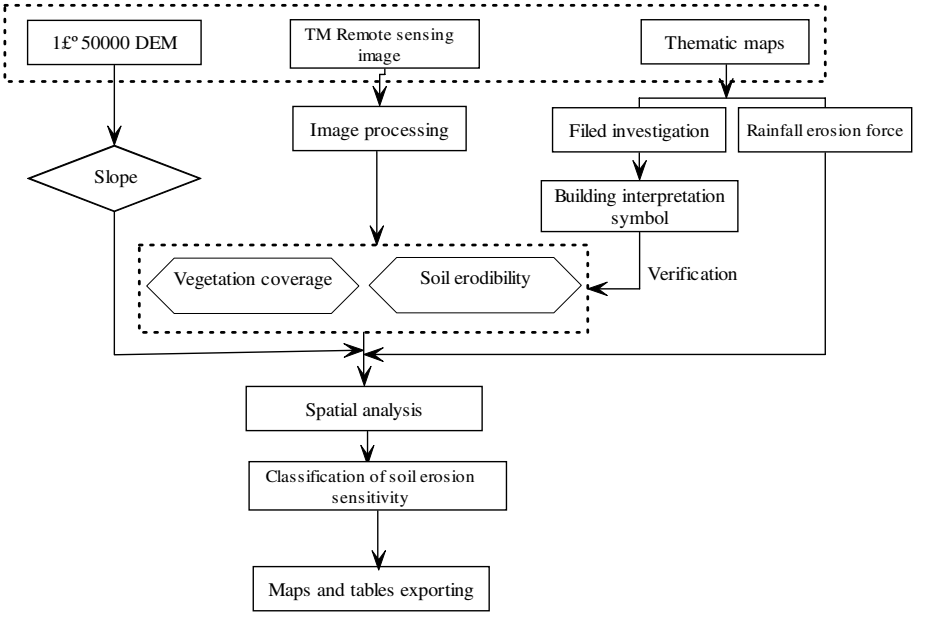


Fig. 1. Technique route for evaluation on sensitivity of soil erosion

4 Evaluation of Soil Erosion Sensitivity

4.1 Single-Factor Sensitivity Evaluation

4.1.1 Rainfall Erosion Force Makes an Impact on Soil Erosion Sensitivity

Rainfall erosion force reflect the corrosive degree of possibility of soil erosion happening under condition of precipitation, the map of corrosive force of precipitation was

graded and reassigned by using Arcview software, which combined map of Rainfall erosion force with administrative map. A result showed that this basin mainly included mild sensitive region and moderate sensitive region, but some areas present high sensitivity.

The area of most sensitive region was 148.88km^2, which account for 0.47% of the total area of this research region, mainly in the northern part of Fei country, middle of research region. The area of high sensitive region was 3837.76km^2, which account for 12.11% of the total area of research region, mainly distributed in middle of Wulian country and Dongang country, and east of Shanting country and Mengyin country, northeast and northwest of Fe country, southwest of Yinan country.

The area of moderate sensitive region was 15626.72km^2, which account for 49.30% of the total area of study region, its area is largest in research region. From geographic distribution we can see that this type traverse the middle of research region form southeast to northwest, mainly distributed in Linshu country and Tancheng country.

The area of mild sensitive region was 11360.84km^2, which account for 35.84% of the total area of this research region. This type was concentrated distribution, mainly in (1) Southeast of research region, which is the zonal distribution from southeast to northwest between moderate sensitive region and insensitive region, the area was 5953.64km^2, accounting for 52.41% of the total area of mild sensitive region, mainly distributed in most part of Zaozhuang city, Tengzhou city, Zhoucheng city, Qufu city, and small part of Weishan country, Sishui country, Yanzhou country, Changshan country and Shanting country. (2) Northeast of research region, which which is the zonal distribution from southeast to northwest, the area was 4465.48 km^2, accounting for 39.31% of mild sensitive region, mainly distributed in Yiyuan country, Yishui country and Ju country. (3) Middle of research region, which present ringy distribution in peripheral region of none sensitivity. The area was 921.92 km^2, accounting for 8.11% of mild sensitive region, mainly distributed in Pingyi country.

4.1.2 Soil Erodibility Makes an Impact on Soil Erosion Sensitivity

Soil is the object and main body for soil erosion. Soil possesses an antierosion capacity to resist the effecting of exogenic force. Soil erodibility is an important index to evaluate the degree of soil erosion sensitivity with quantitative analysis and is an pivotal element to construct the soil erosion forecasting model. The area of 98.52% had different degree of sensitivity, moderate sensitivity was the main type, and the area was 15709.27km2, which account for 49.56% of the total area of research region. The area of moderate and above moderate sensitivity was 29235.2km2, which account for 92.24% of the total area of research region. It maybe concluded that the soil had large possibility for soil erosion in this watershed basin.

4.1.3 Slope Makes an Impact on Soil Erosion Sensitivity

Slope is the important index to evaluate the degree of soil erosion sensitivity and is the important natural factor to cause the happening of soil erosion. According to soil erosion sensitivity distribution map of slope, we can see that none sensitivity was main type, the area of none sensitivity region was 24180.45km^2, which account for 76.29%

of the total area of research region, and most of these areas were plain or depression. The area of mild sensitivity region was 2548.84km^2, which account for 8.04% of total area of the research region, and mainly distributed in piedmont belt of mountainous region, such as piedmont belt of Meng Mountain, Ni Mountain and Wulian Mountain, and so on. Moderate and high sensitivity region were mainly distributed in mountain-side, which is extremely easy to cause soil erosion, these two types account for 7.25% and 6.34% of total area of research area respectively. Most sensitivity region were mainly distributed in upper mountainous region, which account for 2.08% of the total area of research area, and soil erosion was the most severe.

4.1.4 Surface Vegetation Coverage Makes an Impact on Soil Erosion Sensitivity

Vegetation growth and cover condition was effected greatly by human activities, and destruction of the vegetation in some area induced soil erosion, moreover it aggravated degree of soil erosion. Therefore it can reflected the disturbance degree of human activities effect on soil erosion if we made surface vegetation coverage as index to evaluate soil erosion sensitivity. We could get soil erosion sensitivity map of surface vegetation coverage and statistical data of each country through calculation by Arcview software. According to these data, we can see that high sensitivity was main type, which account for 63.95% of total area of research area. The area of moderate sensitivity was 6378.01km^2, account for 20.12% of total area of research area. It maybe concluded that soil erosion sensitivity was higher under effect of surface vegetation cover, and surface vegetation cover was potential factor for severe soil erosion.

4.2 Synthetic Evaluation of Soil Erosion Sensitivity

There was specific process for evaluation of ecological sensitivity based on the grid unit as follows, we made algebraic operation in according with formula (1) for factors of evaluation based on GRID module of ArcGIS software, the result of the calculation showed the sensitive index of soil erosion, the mean value was 4.86. In order to analyze easily, the degree of soil erosion sensitivity was classified five level based on synthetic sensitivity index in Yishusi watershed area.

The type of over mild sensitivity account for 93.17% of total area of research area; the above result showed that there were different degrees of soil erosion sensitivity. The

Table 1. The classification of comprehensive index

Classification of sensitivity	Evaluation of sensitivity	Comprehensive index of sensitivity	Area (km^2)	Proportion (%)
1	None sensitivity	0.0-3.5	2164.49	6.83
2	Mild sensitivity	3.5-4.5	7706.39	24.31
3	Moderate sensitivity	4.5-5.5	16836.39	53.12
4	High sensitivity	5.5-6.5	3733.32	11.78
5	Extremity sensitivity	6.5-10.0	1254.56	3.96

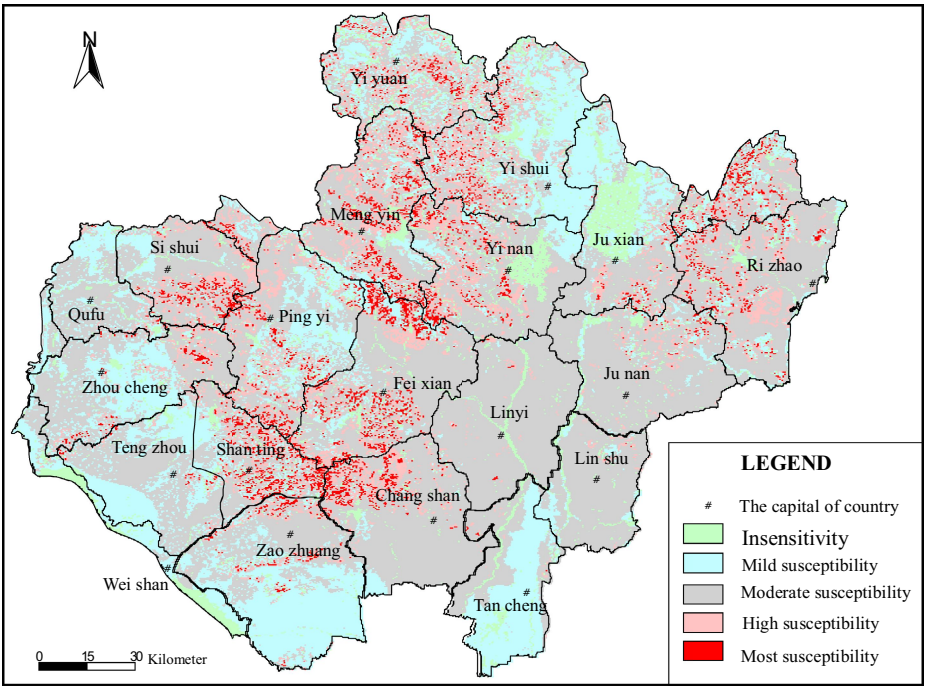


Fig. 2. Distribution map of comprehensive index of soil erosion sensitivity in Yishusi River watershed area

area of mild and over mild sensitivity was 21824.27.17km^2, and account for 68.86% of total area of research region.

The area of insensitive region was 2164.49km^2 in Yishusi watershed area, which account for 6.83% of total area of research area, mainly distributed in Weishan country, Ju country and Yinan country. Mild sensitivity region was secondlarge of area in this watershed area, which area was 7706.39km^2, and account for 24.31% of total area of research region, mainly distributed in Zaozhuang city, Yisui country,tengzhou country and Zhoucheng country. Moderate sensitivity region was first large of area in this watershed area, which area was 16836.39km^2, and account for 53.12% of total area, mainly distributed in mountainous and hilly area. The area of high sensitivity region was 3733.32km^2, which account for 11.78% of total area, mainly distributed in mountain with larger slope. The analysis showed that moderate sensitivity was main type of soil erosion sensitivity.

5 Conclusion

Overall, Yishusi River watershed area of Shandong province is the environmental complex which is composed by compounding of various environmental function areas, but also its ecological environmental is more fragile and more sensitive. Through

evaluation of soil erosion sensitivity, we would grasp the comprehensive information about present situation of soil erosion sensitivity, and provide scientific basis for planning of water and soil conservation and comprehensive management of small watershed area. Based on module of evaluation and analysis, we can evaluate the current situation and quality of elements of soil erosion sensitivity. Final results reflect the temporal and spatial distribution of soil erosion, at the same time it also reflect dynamic variation of soil erosion. This evaluation method not only is fast and accurate, but also timely and workable.

References

1. Ouyang, Z.Y., Wang, X.K., Miao, H.: China's eco-environmental sensitivity and its spatial heterogeneity. Acta Ecologica Sinica 20(1), 9–11 (2000)
2. Sun, X.H.: Research on Soil Erosion Potential Danger in Jinan Based on GIS. Journal of Soil Water Conservation 17(6), 47–50 (2003)
3. Zhou, Y., Zhang, Z.: Risk Assessment of Eco-environment Factors on Soil Erosion Based on GIS. Bulletin of Soil and Water Conservation 22(1), 48–50 (2002)
4. Ma, X.W., Yang, Q.K.: A study on Indexes Choice and Extraction of China Potential Soil and Water LOSS Based on GIS. Bulletin of soil and water Conservation 21(2), 41–44 (2000)
5. Xu, T.S., Peng, S.K., Yue, C.R.: Evaluation of Soil Erosion Based on GIS in a Small Watershed. Journal of Nanjing Forestry University(Natural Sciences Edition) 26(4), 43–46 (2002)

A New Universal Combinatorial Operation Model with Unequal Weights

Lihua Fu[1] and Huacan He[2]

[1] College of Computer Science, Beijing University of Technology, Beijing, 100124, China
[2] College of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, China
fulh1113@yahoo.com.cn

Abstract. The existing universal combinatorial operation only discusses an ideal state that every factor is with equal weight. But every factor is generally with unequal weigh in practical complex system. This paper gives two kinds of general weighted operators. Based on the operators given, an unequal weights universal combinatorial operation model is proposed.

Keywords: Universal logic, Universal combinatorial operation model, Weighted operator, Unequal weights universal combinatorial operation model.

1 Introduction

As is well known, there is complex relation between every factor of complex system, which may be conflictive or consistent. For dealing reasonably with the complex relations, various aggregation operators have been proposed, which are mostly *T*-norm, *S*-norm or *Mean* operators. However, *T*-norm or *S*-norm, can only handle mutually conflictive relation; on the contrary, *Mean* operators can only handle mutually consistent relation, because they have the following properties.

I1 $0 \le T(x, y) \le \min(x, y)$

I2 $\max(x, y) \le S(x, y) \le 1$

I3 $\min(x, y) \le M(x, y) \le \max(x, y)$

According to the above analysis, it can be seen that the operating scope of these aggregation operators are localized. So these operators can not handle roundly the uncertainties in complex system[1-3]. *Universal Logic*[2], proposed by *Prof. He hua_can,* provides a valid path to resolve this problem. Universal combinatorial operation model is the combinatorial connective of *Universal Logic*, which is a cluster of operators determined by the general correlation coefficient *h* between propositions.

In practical complex system, every factor is generally with unequal weight. But the existing universal combinatorial operation only discusses an ideal state that every factor is with equal weight. This paper will give two kinds of general weighted operators. Based on the operators given, an unequal weights universal combinatorial operation model is proposed.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 599–607, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Universal Combinatorial Operation Model

2.1 Universal Combinatorial Operation Model

Universal Logic is a kind of flexible logic. It considers the continuous changeability of not only the true value of proposition, which is called true value flexibility, but also the relation between propositions, which is called relational flexibility. It puts up two important coefficients: *generalized correlation coefficient* "*h*" and *generalized self-correlation coefficient* "*k*". The flexible change of universal logic operators is based on "*h*" and "*k*". So *Universal Logic* provides new foundation to handle roundly the uncertainties in complex system. Universal combinatorial operation model is the combinatorial connective of *Universal Logic*, which is a cluster of operators determined by the general correlation coefficient *h* between propositions.

In this paper, we will only consider the general correlation coefficient *h*. So zero-order universal combinatorial operation model is defined as:

Definition 1[2]. Zero-order universal combinatorial operation model is the cluster.

$$\begin{aligned} & C^e(x,y,h) \\ & = ite\{\Gamma^e[(x^m + y^m - e^m)^{1/m}] \mid x+y < 2e; \\ & \qquad 1-(\Gamma^{1-e}[(1-x)^m + (1-y)^m - (1-e)^m])^{1/m} \mid x+y > 2e; \\ & \qquad e\} \end{aligned} \tag{1}$$

where $m=(3-4h)/(4h(1-h))$, $h\in[0,1]$, $m\in R$, e is the fixed identity, $e\in[0,1]$.

Remark. The conditional expression ite{β | α; γ } represents that if α is true, then the result is β, otherwise γ. Similarly, ite{β_1 | α_1; β_2 | α_2; γ} = ite{β_1 | α_1; ite{β_2 | α_2; γ}}. And $\Gamma^1[x]$ = ite{1| x>1; 0| x<0 or x is an imaginary number; x}.

The zero-order universal combinatorial operation model is a continuous cluster of combinatorial operators determined by *h* between propositions. In practical application, according to the general correlation between propositions, we can take the corresponding one from the cluster.

2.2 Universal Combinatorial Operation Model on Any Interval [*a*, *b*]

In previous work, the model has been established not on any interval [*a*, *b*] but only on [0, 1]. So [4] has put forward a fractional logic based on continuous interval [*a*, *b*], and has discussed the necessity and probability. The *universal combinatorial operation model* in [4], which is on any interval [*a*, *b*], is defined as:

Definition 2[4]. *Universal combinatorial operation model* in any interval [*a*, *b*] is the cluster.

$$
\begin{aligned}
&GC^{e}(x,y,h)\\
&=ite\{\min\left(e,(b-a)\left[\max\left(0,\frac{(x-a)^{m}+(y-a)^{m}-(e-a)^{m}}{(b-a)^{m}}\right)\right]^{\frac{1}{m}}+a\right)\Big|x+y<2e;\\
&\quad b+a-\min\left(e',(b-a)\left[\max\left(0,\frac{(b-x)^{m}+(b-y)^{m}-(b-e)^{m}}{(b-a)^{m}}\right)\right]^{\frac{1}{m}}+a\right)\Big|x+y>2e;\\
&\quad e\}
\end{aligned}
\tag{2}
$$

where m=(3-4h)/(4h(1-h)), $h\in[0,1]$, $m\in R$, $e,e'\in[a,b]$, $e'=b+a-e$.

3 Unequal Weights Universal Combinatorial Operation Model

In practical complex system, every factor is generally with unequal weight. But the existing universal combinatorial operation only discusses an ideal state that every factor is with equal weight.

3.1 Weighted Operator

For dealing reasonably with the complex relations of every factor in complex system, various properties which weighted operators should have are put forward. The weighted operator proposed by *Yager* is one of the famous ones.

3.1.1 Yager Weighted Operator

The weighted operator proposed by Yager is defined as:

Definition 3[3]. Assume an operator $h(\alpha,x)$ is a mapping from [0, 1] to [0, 1]. $h(\alpha,x)$ is called a *Yager weighted operator* of *Yager aggregation operator* YC^{e} if it satisfies the following properties:

I1: Monotonicity with respect to the value, x. In particular if $x>x'$ then we require

$$h(\alpha,x)\geq h(\alpha,x').$$

I2: We desire that elements with weight zero have no effect on the aggregation process, thus if e is the fixed identity of the aggregation to be used on the resulting bag we must have

$$h(0,x)=e.$$

I3: A normalcy with respect to the weights

$$h(1,x)=x.$$

I4: Finally, we desire that the transformation moves monotonically from its value for α =0 to α =1. That is, if $x \geq e$, $h(\alpha, x)$ increases monotonically with respect to the value α; if $x \leq e$, $h(\alpha, x)$ decreases monotonically with respect to the value α.

where α is the weight associated with an argument, and x is the argument. Both α and x are drawn from [0, 1].

And due to the definition of *Yager weighted operator*, we can draw the following theorems:

Theorem 1. Assume $h(\alpha, x)$ is a *Yager weighted operator* of *Yager aggregation operator* YC^e. So we have:

If $x \geq e$, then $h(\alpha, x) \geq e$, and if $x \leq e$, then $h(\alpha, x) \leq e$.

Proof

Due to the properties I2 and I4 of the definition, the theorem can be proved easily.

Theorem 2. Assume $h(\alpha, x)$ is a *Yager weighted operator* of *Yager aggregation operator* YC^e. So we have:

If $x \geq e$, then $h(\alpha, x) \leq x$, and if $x \leq e$, then $h(\alpha, x) \geq x$.

Proof

Due to the properties I3 and I4 of the definition, the theorem can be proved easily.

The chart of *Yager weighted operator* $h(\alpha, x)$ changing along with the weight α is given in Fig.1.

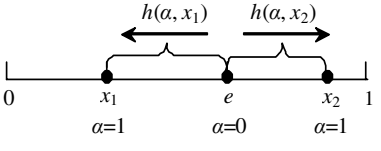


Fig. 1. Chart of changing along with the weight α of *Yager weighted operator* $h(\alpha, x)$

One formulation that satisfies these conditions is $h(\alpha, x) = \alpha x + (1-\alpha)e$. It is described by Fig.2.

3.1.2 General Weighted Operators

According to the above analyses, we discover that Yager weighted operator has some shortcomings as follows.

1) Yager weighted operator is likely to transform entire true(false) proposition into partial true(false) proposition. However, due to the view of logic, entire true(false) proposition should still be transformed into entire true(false) proposition by weighted operator.

2) The weighted value changes in the interval [e, x] or [x, e] for any weight $\alpha \in [0,1]$ as Fig.1 describes. However, we desire the weighted value can changes in the total interval [0, 1] in some practical applications.

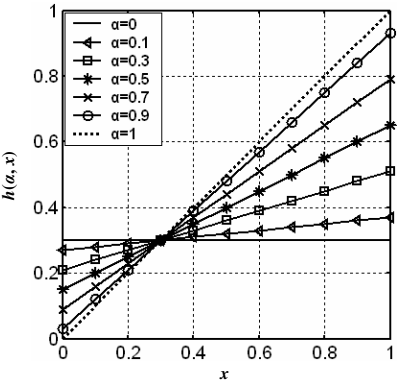


Fig. 2. Chart of the *Yager weighted operator* $h(\alpha, x) = \alpha x + (1-\alpha)e$ with e=0.3

3) *Yager weighted operator* is limited in the interval [0, 1]. But weighted operators are desired to change in the general interval [a, b] in some practical applications.

For solving the above problems, the paper puts forward two kinds of general weighted operators $Gh(\alpha, x)$, which change in the general interval [a, b].

Definition 4. Assume an operator $Gh(\alpha, x)$ is a mapping from [a, b] to [a, b]. $Gh(\alpha, x)$ is called a *general weighted operator I* of *Universal combinatorial operator* GC^e if it satisfies the following properties:

I1: Monotonicity with respect to the value, x. In particular if $x > x'$ then we require

$$Gh(\alpha, x) \geq Gh(\alpha, x').$$

I2: $Gh(0, x) = ite\{a \mid x = a; b \mid x = b; e\}$.

I3: $Gh(\alpha, e) = e$.

I4: $Gh(\alpha, a) = a, Gh(\alpha, b) = b$.

I5: Finally, we desire that the transformation moves monotonically for the weight $\alpha \in (0,1)$. That is, if $x \geq e$, $Gh(\alpha, x)$ increases monotonically with respect to the value α; if $x \leq e$, $Gh(\alpha, x)$ decreases monotonically with respect to the value α.

I6: $Gh(1, x) = x$.

where α is the weight associated with an argument x, and e is the fixed identity of the aggregation operator GC^e. α is drawn from [0, 1] and e is drawn from [a, b].

According to the above definition, we can know that the absolute value of the argument x always decreases with the weight α changing continuously from 0 to 1. That is consistent with *Yager weighted operator* as Fig.3 describes.

Definition 5. Assume an operator $Gh(\alpha, x)$ is a mapping from [a, b] to [a, b]. $Gh(\alpha, x)$ is called a *general weighted operator II* of *Universal combinatorial operator* GC^e if it satisfies the following properties:

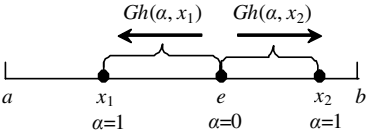


Fig. 3. Chart of changing along with the weight α of *general weighted operator I*

I1: Monotonicity with respect to the value, *x*. In particular if $x > x'$ then we require

$$Gh(\alpha, x) \geq Gh(\alpha, x').$$

I2: $Gh(0, x) = ite\{a \mid x = a; b \mid x = b; e\}$.

I3: $Gh(\alpha, e) = e$.

I4: $Gh(\alpha, a) = a, Gh(\alpha, b) = b$.

I5: Finally, we desire that the transformation moves monotonically for the weight $\alpha \in (0,1)$. That is, if $x \geq e$, $Gh(\alpha, x)$ increases monotonically with respect to the value α; if $x \leq e$, $Gh(\alpha, x)$ decreases monotonically with respect to the value α.

I6: $Gh(1, x) = ite\{e \mid x = e; a \mid x < e; b \mid x > e\}$.

where α is the weight associated with an argument *x*, and *e* is the fixed identity of the aggregation operator GC^e. α is drawn from [0, 1] and *e* is drawn from [*a*, *b*].

According to the above definition, we can know that the absolute value of the argument *x* decreases first and then increases with the weight α changing continuously from 0 to 1. The chart of *general weighted operator II* changing along with the weight α is given in Fig.4.

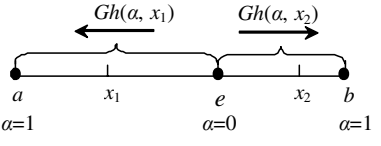


Fig. 4. Chart of changing along with the weight α of *general weighted operator II*

The common general weighted operation models are Polynomial model and Exponential model.

1. *Polynomial model*

$$Gh(\alpha, x) = ite\{\frac{(x-e)(b-e)(1+n)^{1/2}}{((b-x)+(1+n)^{1/2}(x-e))} + e \mid x > e;$$
$$e - \frac{(e-x)(e-a)(1+n)^{1/2}}{((x-a)+(1+n)^{1/2}(e-x))} \mid x < e; \quad (6)$$
$$e\}$$

1) Assume that $n=(4(\alpha-1))/(2-\alpha)^2$, $\alpha\in[0,1]$, $n\in[-1,0]$, and $Gh(\alpha,x)$ is the limit as α = 0. So Eq.(6) describes a *general weighted operator I* as a) of Fig.5 shows.

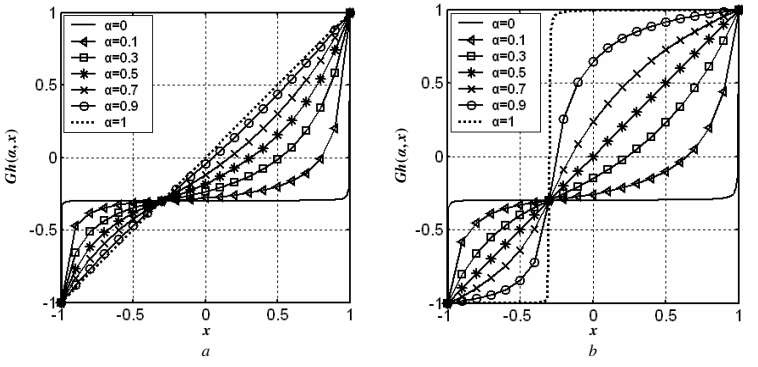


Fig. 5. Chart of polynomial model a) general weighted operator I; b) general weighted operator II with a = -1, b = 1, e = -0.3

2) Assume that $n=(2\alpha-1)/(1-\alpha)^2$, $\alpha\in[0,1]$, $n\geq-1$, and $Gh(\alpha,x)$ is the limit as α = 0 and α = 1. So Eq.(6) describes a *general weighted operator II* as b) of Fig.6 shows.

2. *Exponential model*

$$Gh(\alpha,x)=ite\{(b-e)((x-e)/(b-e))^n+e \mid x>e;\\ e-(e-a)((e-x)/(e-a))^n \mid x<e;\\ e\} \tag{8}$$

1) Assume that $n=\ln 2/(\ln 2-\ln(2-\alpha))$, $\alpha\in[0,1]$, $n\geq 1$, and $Gh(\alpha,x)$ is the limit as α = 0. So Eq.(8) describes a *general weighted operator I* as Fig.6 shows.

2) Assume that $n=-\ln 2/\ln(1-\alpha)$, $\alpha\in[0,1]$, $n>0$, and $Gh(\alpha,x)$ is the limit as α = 0 and α = 1. So Eq.(8) describes a *general weighted operator II* as Fig.7 shows.

From the above curve charts of the two common models changing along with the weight α, we can find the commonness and little difference between them. Polynomial model has good properties and simple computation. So it is a very perfect weighted operator model and generally used in practical applications. However, from the research on circuit implementation of universal logics operations, it is found that the exponent model can be realized more easily by physical apparatus. Therefore each model has its advantage.

3.2 Unequal Weights Universal Combinatorial Operation Model

According to the above definition of general weighted operators, we can get the definition of *Unequal weights universal combinatorial operation model* as follows.

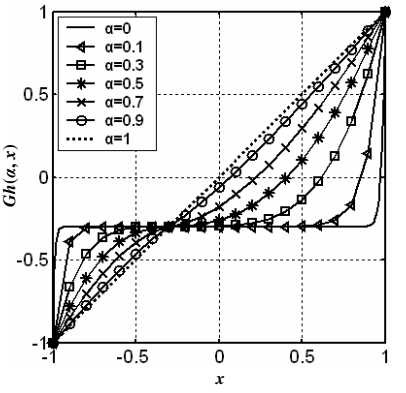


Fig. 6. Chart of exponential model of general weighted operator I with a = -1, b = 1, e = -0.3

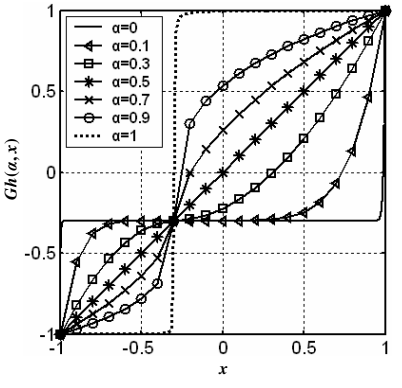


Fig. 7. Chart of exponential model of general weighted operator II with a = -1, b = 1, e = -0.3

Definition 6. Assume an operator $UGC^e(x, y, \alpha_x, \alpha_y, h)$ is a mapping from [*a*, *b*] to [*a*, *b*]. $UGC^e(x, y, \alpha_x, \alpha_y, h)$ is called *unequal weights universal combinatorial operation model*:

$$UGC^e(x, y, \alpha_x, \alpha_y, h) = GC^e(Gh(\alpha_x, x), Gh(\alpha_y, y), h) \tag{10}$$

where $Gh(\alpha, x)$ is *general weighted operator*, $GC^e(x, y, h)$ is *universal combinatorial operator*, *e* is the fixed identity of $GC^e(x, y, h)$, *h* is the *generalized correlation coefficient*, α_x and α_y denote respectively the weights associated with the arguments *x* and *y*. Both *x*, *y* and *e* are drawn from [*a*, *b*], and *h*, α_x and α_y are drawn from [0, 1].

The chart of *unequal weights universal combinatorial operation model* is given in Fig.8.

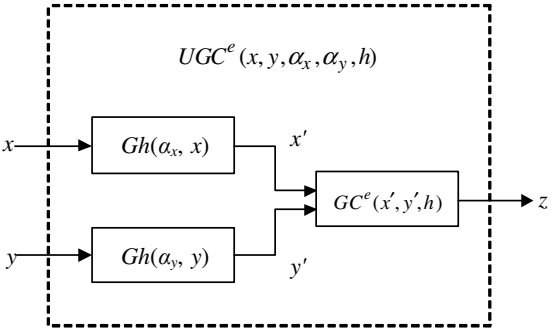


Fig. 8. Chart of unequal weights universal combinatorial operation model

4 Conclusions

In practical complex system, every factor is generally with unequal weight. But the existing universal combinatorial operation only discusses an ideal state that every factor is with equal weight. So the paper has given two kinds of general weighted operators. Based on the operators given, an unequal weights universal combinatorial operation model is proposed. So it is desired that we can deal more reasonably with the complex relations between every factor of complex system.

Acknowledgments. This work is supported by the ph.D Foundation (52007012200702), Young Teacher Foundation(97007012200701) in Beijing University of Technology, Subject and Graduate Education Construction in Beijing Municipal Commission of Education , China.

References

1. Fu, L.H.: Studies on theory and application of flexible logic control for complex system[D]. Northwestern polytechnical university, Xi'an (2005)
2. He, H.C., Wang, H., Liu, Y.H., et al.: Universal Logics Principle [M]. Science press, Beijing (2001)
3. Yager, R.R.: Full reinforcement operators in aggregation techniques. IEEE Transaction on Systems, Man, and Cybernetics 28(6), 757–769 (1998)
4. Chen, Z.C.: Studies on Correlation Reasoning of Fractal, Chaos, and Logic in Complex System [D]. Northwestern polytechnical university, Xi'an (2004)
5. Li, S.Y.: Fuzzy Control, Neurocontrol and Intelligent Cybernetics [M]. Harbin institute of technology press, Harbin (1996)

An Observer-Based Neural Network Controller for Chaotic Lorenz System

Suwat Kuntanapreeda

Research and Development Center for Intelligent Systems
Faculty of Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand 10800
suwat@kmutnb.ac.th

Abstract. The paper proposes an alternative control design for the chaotic Lorenz system based on neural networks. The controller is a feedforward neural network trained by a model reference technique. Implementation of the control design requires system states for feedback, while in most of practical applications only the system output is available. To overcome this problem, a nonlinear observer is used to estimate the states of the system. Simulation results have illustrated the feasibility and effectiveness of the proposed observer-based neural network controller.

Keywords: Neural network, Neural network control, Observer-based control, Chaos control, Lorenz system, Thau observer.

1 Introduction

Chaotic systems are very interesting dynamical systems which have been intensively studied during the last two decades. Although the chaotic systems are deterministic systems, their behavior is very sensitive to initial conditions and unpredictable. Because of the difficulty of accurate prediction of a chaotic system behavior, chaos may cause system instability or degradation in performance. In the field of chaos control, the Lorenz system is often taken as a paradigm, since it captures many of the features of chaotic dynamics. Many methods have been developed for controlling of chaos such as O-G-Y method [1], backstepping design [2-4], sliding mode control [5-6], neural network control [5,7], fuzzy logic control [8] and many others [9-10].

The most popular nonlinear control design for controlling chaos includes sliding model control and backstepping design. The sliding mode control provides an effective alternative to deal with uncertain chaotic systems. However, it results an undesirable chattering phenomenon due to finite-time switching delay. This will deteriorate system performance and also cause wear and tear in mechanical devices. The backstepping design is capable of generating a globally asymptotically stabilizing control laws. The design is based on the Lyapunov stability theory. However, the design results a very complicated control law, which is not easy for implementation. Nowadays, intelligent control based on neural networks and fuzzy logic has become

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 608–617, 2008.
© Springer-Verlag Berlin Heidelberg 2008

attractive solutions for nonlinear control problems. This is because the control design is based on training processes.

This paper presents an alternative control design for the Lorenz system based on neural networks. The controller is a multi-layer feedforward neural network trained by a model reference technique [11,12]. Since the controller requires all of the system states for computing the control command, the Thau nonlinear observer [2,13-14] is used to estimate the states of the system. Simulation results verify that the proposed observer-based neural network controller is an alternative effective controller for controlling chaos.

2 Controller Design

2.1 Multi-layer Feedforward Neural Networks

Neural networks are a promising new generation of information processing systems that demonstrate the ability to learn from training data. This concept of trainable networks further strengthens the idea of utilizing the learning ability of neural networks to learn parameters of control or decision systems. Multi-layer feedforward neural networks are one of the most extensively used neural networks in control problems. They consist of an input layer, hidden layers, and an output layer. For a single hidden layer network, the output of the hidden layer is

$$h = \sigma\left(V^T x + a\right)$$

where

$V = \lfloor V_{ji} \rfloor$ is the weight matrix

$a = [a_1, a_2, ...]$ is the bias vector

$x = [x_1, x_2, ...]$ is the input vector

$\sigma(.)$ is the activation function.

The output of the network is

$$y = \sigma\left(W^T h + b\right)$$

where

$W = \lfloor W_{ji} \rfloor$ is the weight matrix

$b = [b_1, b_2, ...]$ is the bias vector

$h = [h_1, h_2, ...]$, which is the output vector of the hidden layer, is the input vector of the output layer.

$\sigma(.)$ is the activation function.

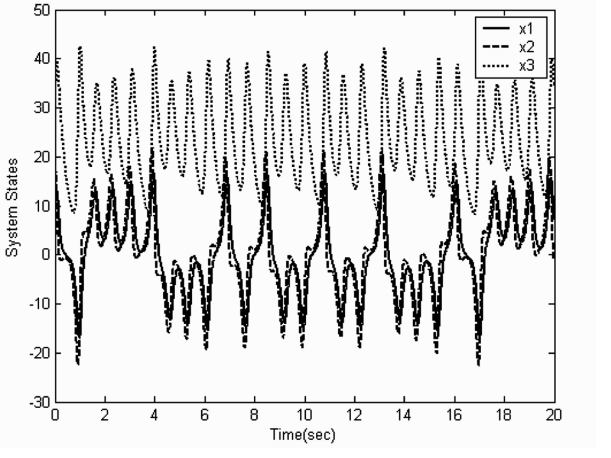
The learning parameters of the network are V, W, a and b. The standard learning rule for the multi-layer feedforward neural networks is backpropagation, which is a gradient descent algorithm. For more details of the neural networks and backpropagation training rule, the reader is referred Ref [15, 16].

2.2 Lorenz System

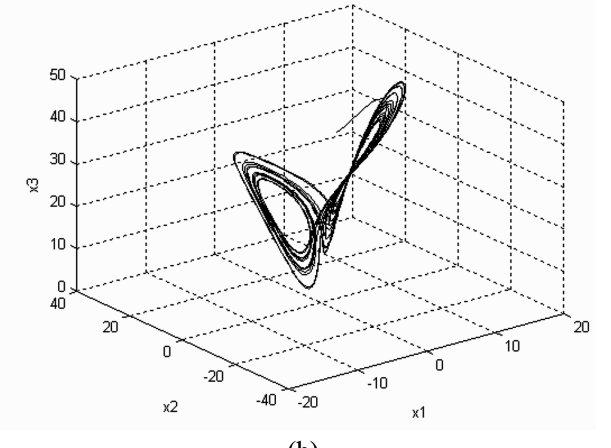
Lorenz system is one of paradigms in exploration of chaos and is described by

$$\begin{aligned} \dot{x}_1 &= p_1(x_2 - x_1) \\ \dot{x}_2 &= p_2 x_1 - x_2 - x_1 x_3 + u \\ \dot{x}_3 &= x_1 x_2 - p_3 x_3 \end{aligned} \tag{1}$$

where x_1, x_2 and x_3 are system state, u is a control input, and p_1, p_2 and p_3 are positive constant parameters. When $p_1 = 10$, $p_2 = 28$ and $p_3 = \frac{8}{3}$, the uncontrolled system presents chaos. The behavior of the uncontrolled system is shown in Fig. 1.



(a)



(b)

Fig. 1. Chaotic behavior of uncontrolled Lorenz system. (a) State response and (b) Lorenz attractor. The initial conditions are $x_1(0) = 10$, $x_2(0) = 20$ and $x_3(0) = 30$.

2.3 Controller

In this paper the control objective is to stabilize the system to the zero state. Let the control input $u = u_1 + u_2$ and $u_1 = x_1 x_3$ such that the system is decoupled as

$$\begin{aligned} \dot{x}_1 &= p_1(x_2 - x_1) \\ \dot{x}_2 &= p_2 x_1 - x_2 + u_2 \end{aligned} \tag{2}$$

and

$$\dot{x}_3 = x_1 x_2 - p_3 x_3 . \tag{3}$$

Eq. (3) shows that x_3 represents the internal dynamic of the system, which will converge to zero when x_1 and x_2 converge to zero. Thus, our control problem is simplified to the stabilization problem of Eq.(2).

A multi-layer feedforward neural network controller is used to generate the control signal u_2. A model reference technique is used to train the controller. The training process is divided into two steps. First, a neural network emulator is trained to adequately duplicate the dynamics behavior of the system to be controlled. After that, the trained emulator is employed to train the controller by allowing the error between the outputs of the control system and reference model to be backpropagated to the output of the controller. This backpropagated error is then used to adjust the weights of the controller. For more details, the reader is referred Ref [11, 12].

2.4 Thau Nonlinear Observer

The above proposed controller requires all of the system states, x_1, x_2 and x_3 to compute the control input, u. However, in practice, not all of the states is usually accessible. In this paper, we assume that x_1 is the system output and is the only accessible state. Therefore, an appropriate state observer is needed to estimate the remaining states. Here, Thau nonlinear observer will be employed. The observer is written as

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + f(\hat{x}) + Bu + K(y - \hat{y}) \\ \hat{y} &= C\hat{x} \end{aligned} \tag{4}$$

where A, B, C and $f(.)$ are known matrices and function of the system to be observed and y is the accessible output of the system. The function $f(.)$ is required to be Lipschitz. Here, K is the observer gain. If K is selected such that the eigenvalues of $A_0 = A - KC$ are in the left half-plane and the inequality

$$L < \frac{\lambda_{\min}(Q)}{\|P\|} \tag{5}$$

where

L is the Lipschitz constant of the function $f(.)$

P, Q are the positive definite matrices that satisfy the following Lyapunov equation:

$$A_0^T P + PA_0 = -2Q\,,$$

is satisfied, then the estimation state $\hat{x}$ is asymptotically converge to the system state x. For more details, the reader is referred Ref [13]. A simpler version of Eq (5) is given as [14]

$$2L < \lambda_{\min}(-S), \quad S = A_0 + A_0^T\,. \tag{6}$$

3 Simulation Results

In this section, the effectiveness of the proposed observer-based controller is verified via computer simulations.

First, one thousand input-output data points randomly sampled from Eq.(2) are used to train the neural network emulator. A sampling period of 0.01 second is used throughout. The emulator is configured as a single hidden layer network; with three inputs to the input layer, followed by a ten-neuron hidden layer, and an output layer with two neurons. The training is terminated after the sum-square-error between the outputs of Eq.(2) and emulator is less than 0.0001. For training the controller, the emulator is no further adjusted. A reference model is selected corresponding to a second order linear system with the damping ratio of 0.707 and the settling time of 1 second. Another one thousand input-output data points randomly sampled from the reference model is used to train the controller. A single-hidden layer configuration is also selected for the controller; with two inputs to the input layer, followed by a ten-neuron hidden layer, and an output layer with a single neuron. The training of the controller is terminated after the sum-square-error between the outputs of the reference model and control system is also less than 0.0001.

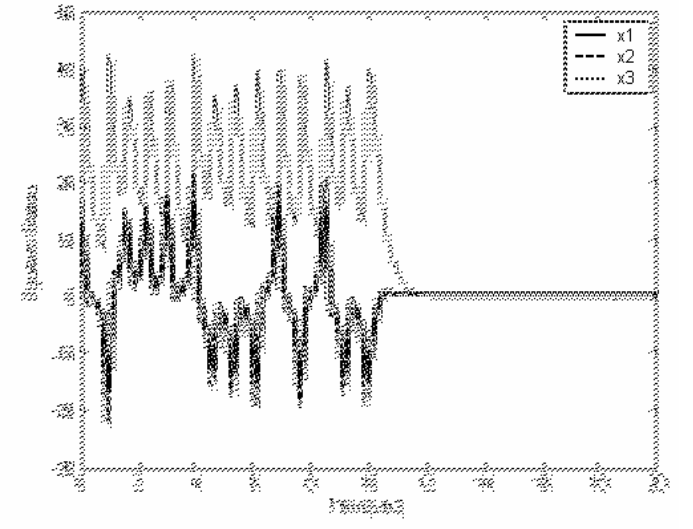
For the Thau observer, the design result from Ref [2] is employed here. The observer gain $K = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}^T = \begin{bmatrix} 10 & 28 & 0 \end{bmatrix}^T$ is used. The equations for the observer, which are rescaled to meet the inequality of Eq (6), are

$$\begin{aligned}
\dot{\hat{w}}_1 &= p_1(\hat{w}_2 - \hat{w}_1) + k_1(w_1 - \hat{w}_1) \\
\dot{\hat{w}}_2 &= p_2\hat{w}_1 - \hat{w}_2 - \mu\hat{w}_1\hat{w}_3 + \frac{u}{\mu} + k_2(w_1 - \hat{w}_1) \\
\dot{\hat{w}}_3 &= \mu\hat{w}_1\hat{w}_2 - p_3\hat{w}_3 + k_3(w_1 - \hat{w}_1)
\end{aligned} \tag{7}$$

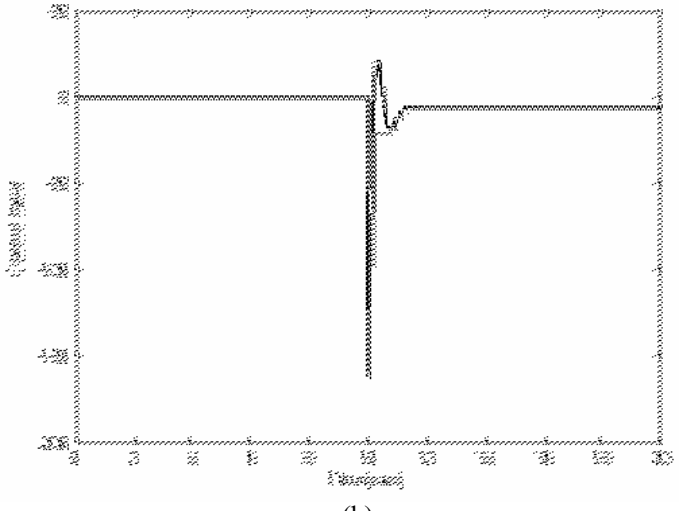
where $w_1 = x_1/\mu$ and $\hat{x} = \mu w$.

Fig. 2 shows the results of the proposed observer-based controller. The initial estimates of the states are set to zero. The initial condition $x_1(0) = 10$, $x_2(0) = 20$ and $x_3(0) = 30$ are used in all simulations. The controller is activated at t = 10 sec. The results illustrate that the controller is able to stabilize the system as desired.

Moreover, in order to exhibit the robustness of the control system, we also consider the scenarios in which the Lorenz system is externally disturbed and the parameters of



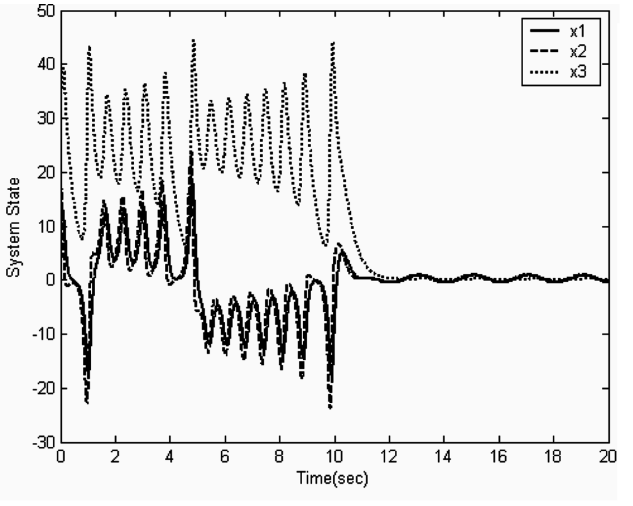
(a)



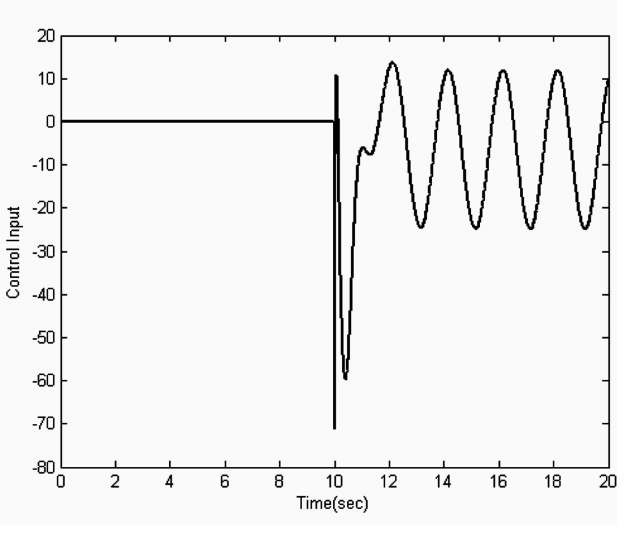
(b)

Fig. 2. System responses of the controlled Lorenz system. (a) State response and (b) Control input. The controller is activated at t = 10 seconds.

the system vary from their nominal values. In the first scenario the disturbance $d(t)$ is added to the second equation of Eq. (1). The system responses are shown in Fig. 3. Here, the disturbance is selected as $d(t) = 0.5\sin(\pi t)$. The results illustrate that the controller has capability to stabilize the disturbed Lorenz system as well.



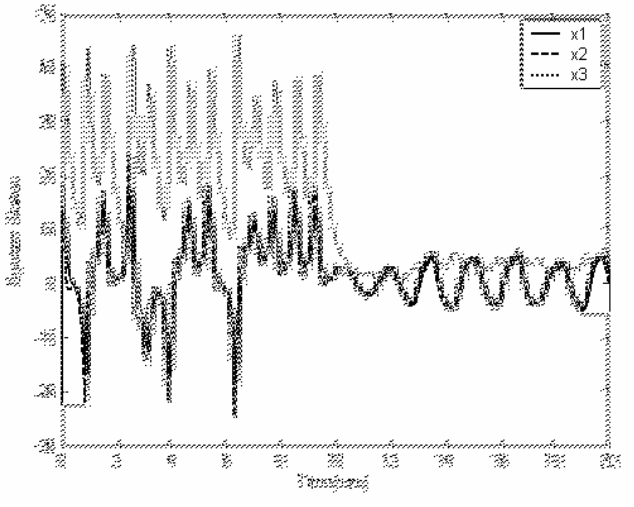
(a)



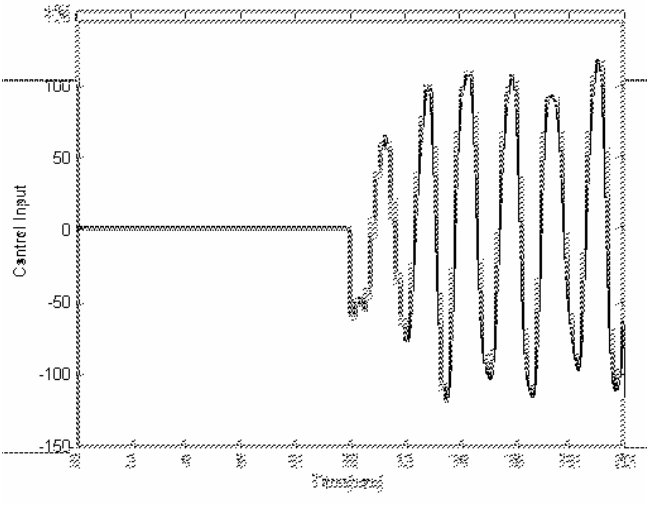
(b)

Fig. 3. System responses of the controlled Lorenz system with external disturbance. (a) State response and (b) Control input. The controller is activated at t = 10 seconds.

For the parameter variation scenario, we let $p_1 = 10 + \sin(2t)$, $p_2 = 28 + 2\cos(8t)$ and $p_3 = \frac{8}{3} + \frac{\cos(5t)}{3}$. Fig. 4 depicts the effectiveness of the control performance. Fig. 5 shows the control responses when both disturbance and parameter variation occur in the same time.

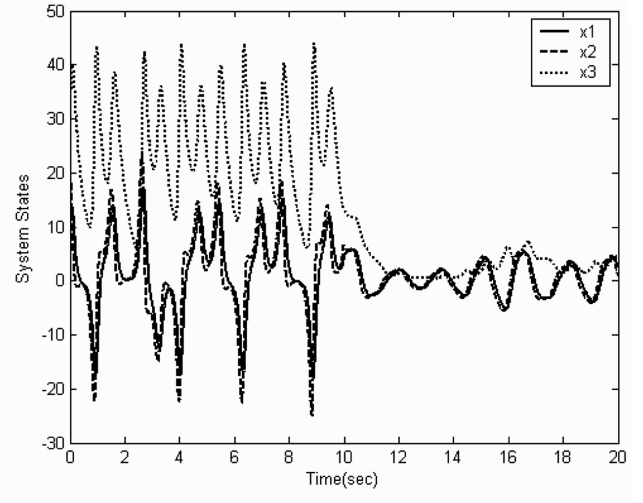


(a)

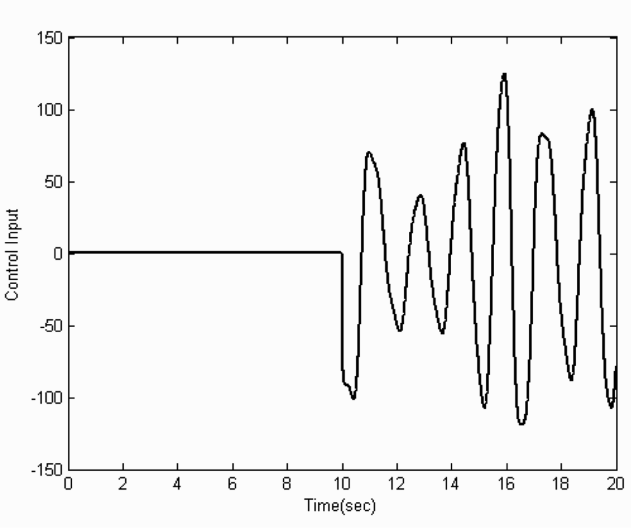


(b)

Fig. 4. System responses of the controlled Lorenz system with parameter variation. (a) State response and (b) Control input. The controller is activated at t = 10 seconds.



(a)



(b)

Fig. 5. System responses of the controlled Lorenz system with external disturbance and parameter variation. (a) State response and (b) Control input. The controller is activated at t = 10 seconds.

4 Conclusions

In this paper, an observer-based neural network control design has been proposed as an alternative solution for controlling the chaotic Lorenz system. The neural network controller is a feedforward network trained by a model reference technique. A Thau

nonlinear observer is used to estimate the states of the system for feedback purpose. Computer simulations show that the proposed solution is effective.

References

1. Ott, E.F., Grebogi, C., Yorke, J.A.: Controlling Chaos. Phys. Rev. Lett. 64, 1196–1199 (1990)
2. Mahboobi, S.H., Shahrokhi, M., Pishkenari, H.N.: Observer-based Control Design for Three Well-known Chaotic Systems. Chaos, Solitons and Fractals 29, 381–392 (2006)
3. Peng, C.-C., Chen, C.-L.: Robust Chaotic Control of Lorenz System by Backstepping Design. Chaos, Solitons and Fractals 37, 598–608 (2008)
4. Chen, F., Chen, L., Zhang, W.: Stabilization of Parameters Perturbation Chaotic System via Adaptive Backstepping Technique. Applied Mathematics and Computation 200, 101–109 (2008)
5. Guo, H., Lin, S., Liu, J.: A Radial Basis Function Sliding Mode Controller for Chaotic Lorenz System. Physics Letters A 351, 257–261 (2006)
6. Nazzal, J.M., Natsheh, A.N.: Chaos Control using Sliding-mode Theory. Chaos, Solitons and Fractals 33, 695–702 (2007)
7. Shen, L., Wang, M.: Adaptive Control of Chaotic Systems Based on a Single Layer Neural Network. Physics Letters A 368, 379–382 (2007)
8. Yau, H.-T., Shieh, C.-S.: Chaos Synchronization using Fuzzy Logic Controller. Nonlinear Analysis: Real World Applications 9, 1800–1810 (2008)
9. Suykens, J.A.K., Curran, P.F., Vandewalle, J.: Robust Nonlinear H-inf Syschonization of Chaotic Lur's Systems. IEEE Trans. Circuits Syst. I. 44, 891–904 (1997)
10. Gambino, G., Lombardo, M.C., Sammartino, M.: Global Linear Feedback Control for the Generalized Lorenz System. Chaos, Solitons and Fractals 29, 829–837 (2006)
11. White, D.A., Sofge, D.A.: Handbook of Intelligent Control. Van Nostrand Reinhold, New York (1992)
12. Kuntanapreeda, S., Gundersen, R.W., Fullmer, R.R.: Neural Network Model refernce Control of Nonlinear Systems. In: IJCNN International Joint Conference on Neural Networks, Baltimore, Maryland, vol. 2, pp. II94 - II9 (1992)
13. Thau, F.E.: Observing the States of Nonlinear Dynamic Systems. Int. L. Control 17, 471–479 (1973)
14. Starkov, K., Esquer, M.: Construction of Observers for Nonlinear Systems based on the Solution of Thau Inequality. In: 14th ICSE Conference, Coventry, UK, vol. 2, pp. 508–512 (2000)
15. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall International, New Jersey (1999)
16. Wasserman, P.D.: Advanced Methods in Neural Computing. Van Nostrand Reinhold, New York (1993)

Integrated Agent-Based Modeling with GIS for Large Scale Emergency Simulation

Dan Guo, Bo Ren, and Cheng Wang

College of Hydropower and Information Engineering,
Huazhong University of Science and Technology,
430074 Wuhan, China
guodandd@126.com

Abstract. Recent development in the area of complex dynamic systems includes the integration of agent-based modeling (ABM) and Geographical Information Systems (GIS), which is used to simulate spatial dynamic processes. The use of ABM for experimenting and exploring geographical phenomena is still in its infancy. The paper has presented an architecture which integrates ABM with GIS, and developed a generic model for emergency response planning against catastrophes, which applied to an urban firefighting simulation in Foshan city. Through the integration, agent-based models including real-time GIS data feeds to simulate and visualize situations unfolding in real time. It provides an accessible way to model the emergence of phenomena through individual interactions of features and real geographies over time and space.

Keywords: ABM, GIS, Emergency Simulation, Repast.

1 Introduction

It has been long recognized that it was difficult to describe and predict the behavior of a complex dynamic system with analytical approaches, especially for the application domain of human-environment interactions, such as large scale emergency simulation. In recent years, the agent-based approach is of particular interest to social/management scientists because human groups, organizations and societies may also be thought of as agents interacting with each other. Emergence has meaningful orderly patterns from the non-linear interactions between autonomous disaggregated parts (agents) and the landscape. Using agent-based modeling (ABM) within a GIS environment becomes an attractive alternative. ABM enables us to simulate the individual actions of diverse agents and measure the resulting system behaviors and outcomes over time. Unlike cellular automata (cells are identical, and always homogeneous and dense), agents in ABM have diverse, heterogeneous, and dynamic in their attributes and behavioral rules, and GIS is a useful medium for representing model input and output of a geospatial nature while space is not grid-based. Although agent-based models have been developed for a diverse range of applications, the use of ABM for experimenting and exploring geographical phenomena, specifically linking it to GIS is still in its infancy(see Gimblett, 2002; Parker, 2005 for some recent applications)[1],[2]. Current GIS and ABM software do not support each other in a seamless manner, and an integrated platform that would support both is needed.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 618–625, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The paper has proposed an architecture which integrates ABM with GIS. The architecture is easy to define various agents' types, properties, behavioral rules, and is also primarily used for display, scheduling, importing GIS vector data. It has developed a generic model for emergency response planning against catastrophes, and described an urban firefighting simulation in Foshan city, which incorporates detailed real-world environmental data, to simulate agents' behaviors and processes as change and movement conditioned by GIS data representations of space and geography. The paper has emphasized work in dealing with agent based models, but within a spatial context representation focusing on GIS science. The simulation is carried out in Java RePast 3.1, and with other Java based GIS libraries, JTS, OpenMap.

2 Agent in ABM/GIS

ABM is a computational methodology that allows the analysts to create, analyze, and experiment with artificial worlds populated by agents that interact in nontrivial ways. Every participant in a reaction is modeled individually, rather than en masse. Agents act (See Fig. 1.a), interact with each other agents, and react to their changing environment according to a set of behavioral rules derived from an underlying theory for the processes and interactions within a particular system [3].

With communicating topological and spatial relationships, agents' behavior could be implemented by querying the GIS, sending a "try to move" message to the GIS. Then GIS could respond in turn by either moving the agent as requested (and thus updating the GIS database and associated graphical displays), or returning a message to the agent about why the move could not be executed (e.g., the location was already occupied, or the movement was not allowed) [4].Fig. 1.b gives a agent dynamic behavior process at continue scheduling in GIS.

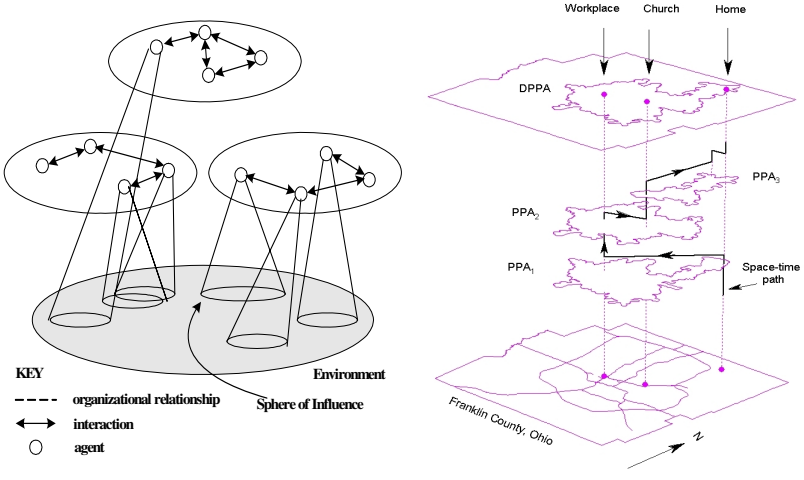


Fig. 1. (a). Agent in ABM. (b). Agent in GIS (Mei-Po Kwan, 2000[5]).

3 An Architecture of ABM-GIS Integration

The paper has proposed an architecture (see Fig. 2), composing of the sophisticated ABM, GIS, and external modules (user interface, auxiliary tools, etc.). The agent-based modeling approach is amenable to incorporate the effect of parameter values, rule-based models and interactions with environment. GIS provides geospatial information to restrict the agents' behavior to within the study area, such as topography, land cover, zoning, transportation, social, information transfer models, etc. External modules are mainly composed of user interface, simulator, visualization, and analysis tools. User interface is in the form of parameter settings via a graphical user interface which allows for the sensitivity testing of model parameters. Simulator sets up and runs external simulation models to carry out all tasks. Visualization and analysis tools are auxiliary means to help performing and researching system outcomes. Results can be exported as different forms to give the model extra spatial analysis functionality. Based this architecture, it could incorporate additional auxiliary tools easily. Just changing the internal parameters or the model of AMB, utilizing the different spatial information, then the AMB-GIS architecture is available in other different application domain of complex systems for human-environment interactions.

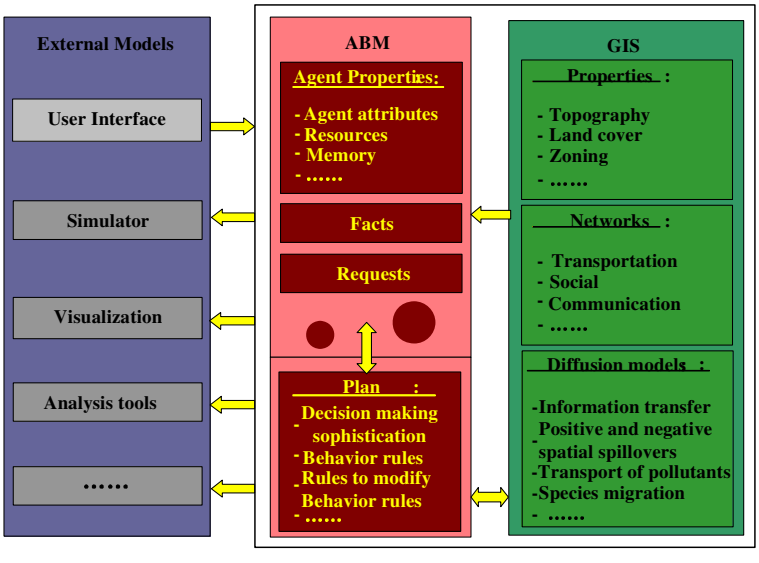


Fig. 2. The architecture of ABM-GIS integration

4 A Generic Model for Emergency Response Planning against Catastrophes

Using above architecture which integrating ABM with GIS functionality, the paper has described how a generic model can be developed and applied to emergency response planning against catastrophes. The model has several parts as follows.

4.1 Agent

Catastrophe: The catastrophe itself is modeled as an agent in the system. It gives the ability to model very different scenarios. In particular, the catastrophe-agent can be specialized in order to model a source of poisoning, a bomb, or a fire, etc. More importantly it is possible to initialize multiple catastrophe-agents and setting their time of activation.

Person: After the catastrophic event, the person agent acts with the environmental and personality factors maintaining information about: (1) the destination (home, work place, hospital or other safe places); (2) current health level, $h \in [0,1]$; (3) degree of worry that represents the innate level of irrationality in the agent, $w \in [0,1]$; (4)level of obedience that captures the instruction-abiding trait of a person, $o \in [0,1]$; and (5) perceived level of distress, $d = w \times(1-h)$; (6) level of learning capability through communication, $c \in [0,1]$.

On-Site responders: On-site treatment is provided by major emergency response vehicles (ambulances, fire engines, etc.), hazardous materials teams and emergency medical services stations. After receiving notification of the disaster, they move towards the catastrophe site. These on-site responders are modeled as dynamic agents in the system, maintaining information about: (1) State: available, critical or full; (2) Facts: location, time of dispatch, resource level (both recoverable resources like emergency response vehicles, and irrecoverable resources like drugs and saline), and reliability of communication device (or rate of information diffusion).

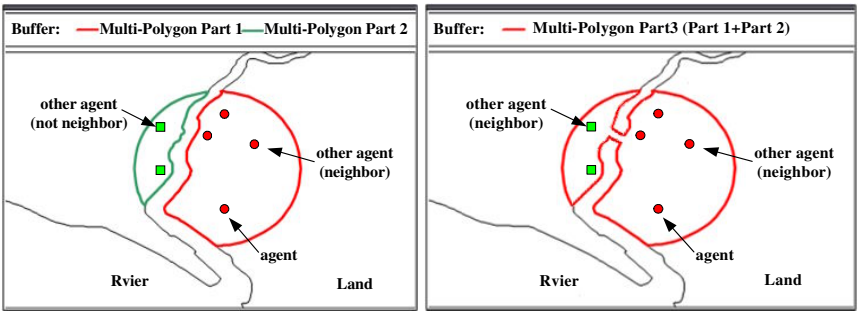
4.2 Topology and Transportation

In this paper, geographic information is taken into account as follow: (a) land use map, (b)walkable environment (future classification-walkability), (c)street network (street segments and nodes), (d)gates (entrances to study area –tube stations, bus stops, boundaries), (e)visibility map (produced from dense set of points). Publicly available GIS data about the roads of the city is converted into a graph, while nodes are intersections and edges are streets. Agents are constrained to move only along the edges of the graph. Thus, agents which type in ABM can be seen as similar to a layer in a GIS, and each agent is similar to a feature in GIS.

A problem linked to using vector space is the treatment of physical boundaries (e.g. rivers and motorways) when calculating neighborhoods. Unlike that of cellular space models where neighborhoods are often calculated using Moore or von Neumann neighborhoods or variations of these, Fig. 3 highlights how geographical features can be incorporated into the model when calculating neighborhoods [6].

4.3 Route Finding

Agent's behavior in GIS is not simply a product of intention and desire in complex systems, but as much a product of uncertainty, randomness and physical constraints of geometry. Based on the rule of route finding, with the environmental and personality factors, the agent decides what to do and where to go.



(a) (b)

Fig. 3. (a). Defining neighborhoods with the inclusion of geographical features (constrained buffer). (B). Defining neighborhoods where the two areas are connected by a bridge.

Behavior rules

Step 1: check position(location)

Step 2: find closer node

Step 3: find node closer to target(destination)

Step 4: load appropriate route finder——shortest path algorithms/utility maximizing paths(composite cost: distance, visibility, active frontage, traffic, walk ability etc.)

Step5: choice decide, then update agents' situation.

The paper has used the LRTA* algorithm (Korf 1990) for route computation to model a person's panic behavior [7]. In order to build model at the appropriate level of description, it could be optimized by modifying the heuristic algorithm which interleaves planning and execution in an on-line decision-making setting.

Rules to modify Behavior rules

With ability to learn and adapt its behavior based on experience, an agent may have rules that modify its rules of behavior. It takes a cost computation to modify behavior rules, formula as follow:

$$C_t = C_e + C_c + C_r \,. \tag{1}$$

At each node, the paper calculates the cost of travel to target through the cost of explorative behavior, communion between agents and each route choice. Ce(the cost of explorative behavior) is calculated with the information, such as the attractiveness of each street segment, visibility, traffic flows, pedestrian flows, etc.; Cc (the cost of communion between agents) is calculated with the information, such as reliability of message, rate of information diffusion, etc.; Cr (the cost of route choice) is the weight of route node in behavior rules.

5 Application

The paper has applied the generic model to an urban firefighting simulation at the gymnasium on the Weiguo road in Foshan city, China. To implement the simulation,

the structure of the program is shown in Fig. 4. The open circles are labeled A, V, D, and P, representing program modules which are called to animate the walks (A), visualize the geometry and surfaces (V), display numeric data (D), and plot such data (P).

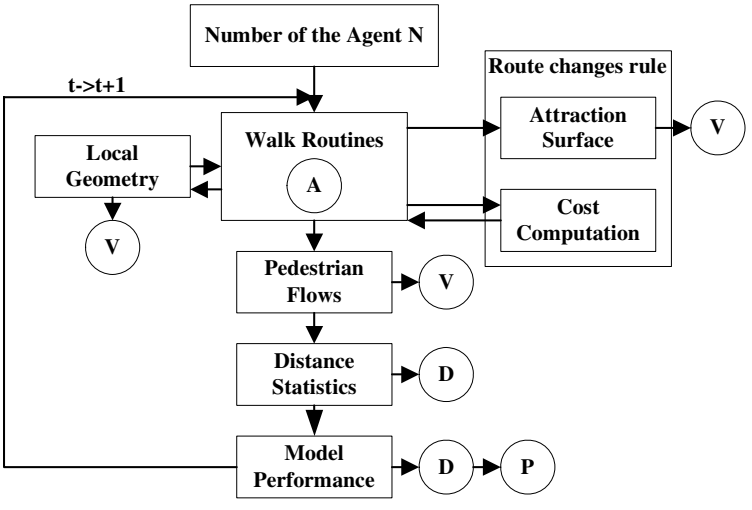


Fig. 4. The structure of integration program

In the simulation system, the streets have been modeled as a non-planar graph. On-site treatment units are assumed to arrive at the scene shortly after the event. The people are modeled as agents initially moving randomly, but with a result ant drift velocity towards their destinations. Upon fire occurrence, people choose to head to one of the safe place or hospitals. The LRTA* algorithm for route computation is used to model a person's panic behavior. Information about safety locations and current capacities is exchanged between adjacent agents, and is also accessible to other agents with some communication devices. The result of the interaction of these 500 agents is analyzed by repeated simulation and parameter sweeps. The maximum number of iterations is 1000. The simulation result at tick count 1575 is seen in Fig. 5.

In order to understand the complexity of the system dynamics, Fig. 6 shows different statistics for the simulation. Within the same scenario, different results are produced by the different population sizes. Fig. 6.A shows the percentage of active agents. As expected, immediately after the catastrophic event, the number of active agents quickly increases, but then decreases. The population size of 1000 agents seems to decreases more slowly. The outcome can be explained by observing that the number of agents with injuries increases with the population size. These are agents who need more treatment producing a longer hospitalization time. It causes more awaiting time. Fig. 6.B shows the evolution curves for the average waiting time of the

affected agents at the hospitals. There is a jump of the curves between 800 ticks and 2600 ticks, as evident in that the population size of 500 agents seems to produce in the higher waiting time at the hospitals. It can be explained by observing that after the nearest hospital becomes full, the remaining waiting agents that heads to another hospital is unable to fill the new one. With the big size population, agents distributing around, they are easy to arrive the new hospital.

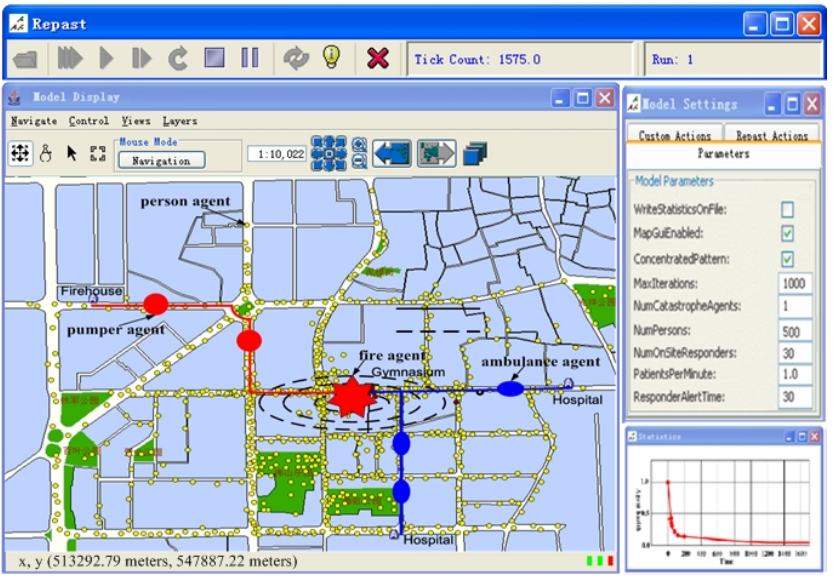
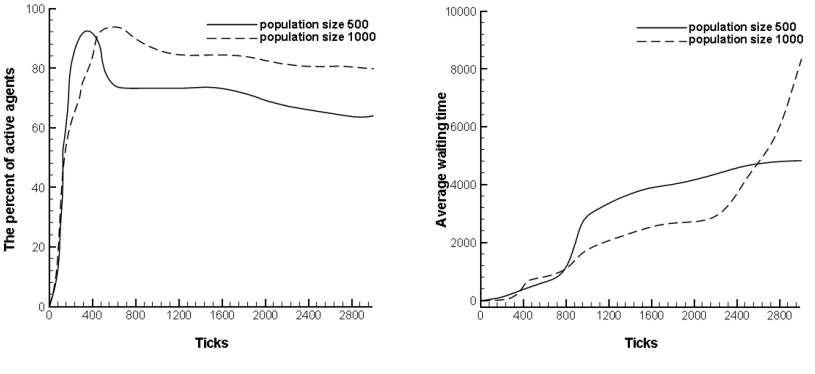


Fig. 5. An urban firefighting simulation at the gymnasium on the Weiguo road in Foshan city, China



(a) (b)

Fig. 6. (a). The average waiting time of the population with population size 500 and 1000. (b). The percentage of active agent with population size 500 and 1000.

6 Conclusions

The paper has presented a generic model which integrates ABM with GIS functionality, to simulate an urban firefighting with detailed real-world environmental data. It is concluded that simulation-based analysis with geospatial information can be used to develop, test, evaluate and refine public health policies governing catastrophe preparedness and emergency response. Currently, ABM used for simulation analysis has only been applied to abstract systems, which was designed to test how space affects the simulation outcome. Further investigation will apply the models to real world situations and to study issues in detail while testing different scenarios. Thus, key criteria pertaining to a selection of simulation / modeling systems, is need to be provided to facilitate the identification of a suitable system for the development of a geospatial agent-based model.

References

1. Gimblett, H.R., Richards, M.T., Itami, R.M.: Simulating Wildland Recreation Use and Conflicting Spatial Interactions using Rule-Driven Intelligent Agents. In: Gimblett, H.R. (ed.) Integrating Geographic Information Systems and Agent-Based Modelling Techniques for Simulating Social and Ecological Processes, pp. 211–243. Oxford University Press, Oxford (2002)
2. Parker, D.C.: Integration of Geographic Information Systems and Agent-Based Models of Land Use: Challenges and Prospects. In: Maguire, D.J., Batty, M., Goodchild, M. (eds.) GIS, Spatial Analysis and Modelling, pp. 403–422. ESRI Press, Redlands (2005)
3. Macal, C.M., North, M.J.: Tutorial on Agent-Based Modelling and Simulation. In: Euhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (eds.) Proceedings of the 2005 Winter Simulation Conference (2005)
4. Brown, D.G., Riolo, R., Robinson, D.T., North, M., Rand, W.: Spatial Process and Data Models: Toward Integration of Agent-Based Models and GIS. Journal of Geographical Systems, Special Issue on Space-Time Information Systems 7(1), 25–47 (2005)
5. Kwan, M.-P.: Gender and Individual Access to Urban Opportunities: A Study Using Space-Time Measures. The Professional Geographer 51(2), 210–227 (1999)
6. Crooks, A.T.: Exploring Cities using Agent Based Models and GIS. In: Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects, Chicago, USA (2006)
7. Korf, R.E.: Real-time heuristic search. Artificial Intelligence 42, 189–211 (1990)

New Research on Harmonics and Reactive Currents Detecting and Its Suppression with ANN in Single-Phase Circuit

Wenjin Dai, Qingsheng Lin, and Yongtao Dai

Department of Electrical and Automatic Engineering,
Nanchang University, Nanchang, Jiangxi, 330031, China
dwj480620@yahoo.com.cn, linqingsheng01@163.com, dytnc2007@126.com

To avoid the deficiencies which exist in the single-phase circuit for detecting harmonics and reactive currents, this paper presents a new detecting method after analysing the operating principle of the single-phase shunt active power filter(SAPF). This new detecting method omits the complex coordinate transform of traditional i_p-i_q detecting method, which can easily obtain sine and cosine signal that in phase of the power supply voltage that do not need PLL circuit. Meanwhile, this new detecting method also proposes the PID negative feedback which can quicken the dynamic response speed. At the same time, this paper also presents a new control method for harmonic and reactive power compensation with an artificial neural network (ANN) controller and a new control algorithm to eliminate harmonics, which makes actual output current of active power filter can accurately track the change of instruction current. Finally, this paper designs a SAPF that compensate harmonics and reactive currents for the single-phase circuit. Through simulation study with Matlab, the simulation results illustrate that the SAPF has a good performance in harmonic suppression and reactive power compensation.

Keywords: harmonic, reactive current, detecting, ANN, SAPF, simulation.

1 Introduction

With the widely use of nonlinear loads in power systems, such as power electronic devices, which has caused an increase of the harmonic disturbances in the power systems[1]. Meanwhile, with the emergence of high-power electronic devices and the development of PWM technology, APF becomes one of the most promising ways of suppressing harmonics, which can dynamicly track compensate harmonics and reactive currents that value and frequency has change[2], [3], [4]. The performance of APF has a lot to the harmonics detecting method and the control method of output current[5]. Therefore, how to detect harmonics and reactive currents of nonlinear loads in real-time seems particularly important. Thus, the current control method directly determine the accuracy and speed of the output of compensation current[6].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 626–636, 2008.
© Springer-Verlag Berlin Heidelberg 2008

At present, most of the APF are three-phase, in fact, the single-phase nonlinear loads (such as electrified railway) also caused serious problems about harmonics, its suppression has become people's concern recently[7]. Now, the main detecting methods that used in APF are Fast Fourier Transform(FFT), detecting method with p-q or ip-iq transform that based on instantaneous reactive power theory and some intelligent detection methods[8], [9], [10], [11]. Among these methods, detecting method with p-q or ip-iq transform that based on instantaneous reactive theory in three-phase circuit is most successful. Thus, extend the idea of three-phase circuit to single-phase circuit, it can acquired by extend another two-phase circuit or shift 90^0 phase of distorted current[12], which is more complex than three-phase harmonics detecting that based on instantaneous theory. The demand of Fast Fourier Transform (FFT) is steady, large amount of calculation ,which result in the real-time is not good[12]; There has a cycle delay of detecting method that based on Fryze, which also have lots of limitations because it can not detecting harmonics or reactive current respectively[12], [13]. In short, there still exist some shortcomings for detecting harmonics and reactive currents real-time in single-phase circuit.

For current control method, SAPF commonly use tracking PWM control circuit because of the high requirement of real-time. At present, there are many intelligent control method to generate PWM pulse to control the switch of SAPF, which makes actual output current of active power filter can accurately track the change of instruction current.

A new method for detecting harmonics and reactive currents in single-phase circuit is presented in this paper, the algorithm of this new method is easy, real-time is also good. For current control method, a new control algorithm with ANN is also presented. Using these methods, a single-phase SAPF for harmonics suppression and reactive power compensation that apply to single-phase circuit is designed. Build simulation model with Matlab software, the simulation results indicate that the new method is feasible.

2 System Constitution and Operation Principle of Single-phase SAPF

Fig.1 shows the system constitution and operation principle of single-phase SAPF, which is mainly composed of harmonics and reactive currents detecting circuit, currents control circuit and APF main circuit. The main APF circuit that presented in this figure is composed of a current control single-phase voltage inverter, which is used to generate compensation current. Each bridge of this inverter is implemented by IGBT and anti-parallel diode. The DC side of inverter is paralleled with capacitor C, which plays an important role in the steady of DC voltage and compensated reactive power. While require compensate harmonics and reactive currents for nonlinear loads, the harmonics and reactive currents are detected by detection circuit from compensated object, then, makes it anti-polarity and as the instruction signal of compensation

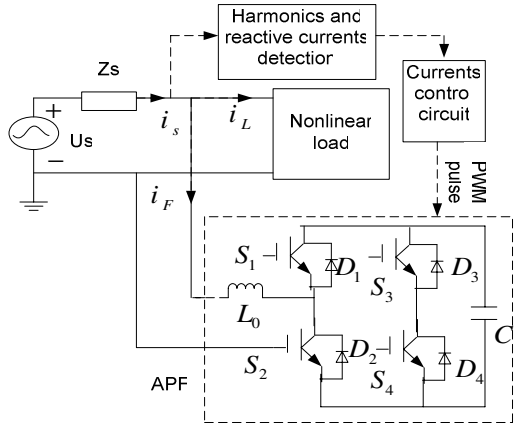


Fig. 1. System constitution of single-phase SAPF

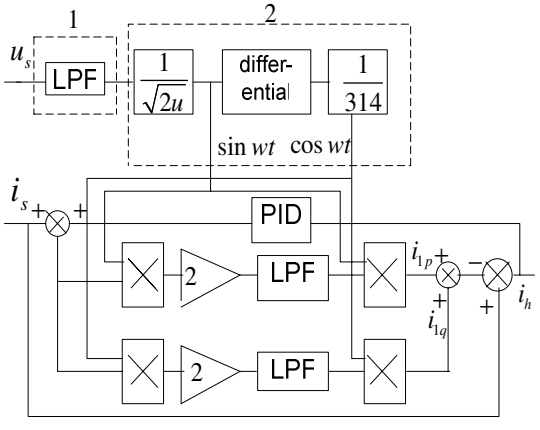


Fig. 2. Detection circuit for harmonics and reactive currents

current, Thus, generate the necessary PWM pulse through control circuit and drive the switch of inverter by some control method. Finally, the inverter generate compensation current.

3 New Detecting Circuit for Harmonics and Reactive Currents

3.1 Basic Detecting Principle and Implement

Supposes the supply voltage u_s is sine wave, $u_s(t) = \sqrt{2}u \sin wt$, where, u is the voltage's effective value and w is angular frequency of power source. The distorted current of nonlinear loads i_L is broken down through Fourier, which can be expressed as follows:

$$\begin{aligned} i_L(t) &= \sum_1^{\infty} \sqrt{2} I_n \sin(nwt + \vartheta_n) \\ &= \sqrt{2} I_1 \sin(wt + \vartheta_1) + \sum_2^{\infty} \sqrt{2} I_n \sin(nwt + \vartheta_n) \\ &= \sqrt{2} I_1 \cos\vartheta_1 \sin wt + \sqrt{2} I_1 \sin\vartheta_1 \cos wt + \sum_2^{\infty} \sqrt{2} I_n \sin(nwt + \vartheta_n) \\ &= i_{1p}(t) + i_{1q}(t) + i_h(t) \end{aligned} \tag{1}$$

Where, I_n, ϑ_n respectively represent the effective value and phase angle of n harmonic, n is a positive integer, whereas, I_1, ϑ_1 respectively represent the effective value and phase angle of fundamental current. As can be seen from Eqs.(1)

$$i_{1p}(t) = \sqrt{2} I_1 \cos\vartheta_1 \sin wt \tag{2}$$

$$i_{1q}(t) = \sqrt{2} I_1 \sin\vartheta_1 \cos wt \tag{3}$$

$$i_h(t) = \sum_2^{\infty} \sqrt{2} I_n \sin(nwt + \vartheta_n) \tag{4}$$

Where, $i_{1p}(t)$, $i_{1q}(t)$ and $i_h(t)$ represent fundamental active current component, fundamental reactive current component and harmonic current component respectively. Define $I_q = \sqrt{2} I_1 \cos\vartheta_1$, $I_q = \sqrt{2} I_1 \sin\vartheta_1$, then, I_p, I_q are expressed as the maximum value of fundamental active and reactive current separately. Therefore, if establish the value of i_p and i_q, we can easily identify fundamental active current $i_{1p}(t)$ and fundamental reactive current $i_{1q}(t)$.

In order to obtain the value of I_p, we can multiply both sides of Eqs.(1) by 2sinwt, which can be expressed as follows:

$$\begin{aligned} 2i_L(t)\sin wt &= 2I_p \sin^2 wt + 2I_q \sin wt \cos wt + 2\sqrt{2} \sum_2^{\infty} I_n \sin(nwt + \vartheta_n) \sin wt \\ &= I_p(1 - \cos 2wt) + I_q \sin 2wt - \sqrt{2} \sum_2^{\infty} I_n \cos((n+1)wt + \vartheta_n) + \sqrt{2} \sum_2^{\infty} I_n \cos((n-1)wt + \vartheta_n) \end{aligned} \tag{5}$$

Obviously, Eqs.(5) is composed of DC component and AC component. So, if multiply $i_L(t)$ by 2sinwt, and then, let it pass through a low pass filter(LPF),we can easily isolate the DC component I_p, which is the maximum value of fundamental active current. Finally, multiply I_p by sinwt, then we can obtain fundamental active current $i_{1p}(t)$。

Similarly, in order to obtain the value of I_q, we can multiply both sides of Eqs.(1) by 2coswt, which can be expressed as follows:

$$\begin{aligned} 2i_L(t)\cos wt &= 2I_p \sin wt \cos wt + 2I_q \cos^2 wt + 2\sqrt{2}\sum_2^\infty I_n \sin(nwt+\vartheta_n)\cos wt \\ &= I_p \sin 2wt + I_q(1+\cos 2wt) + \sqrt{2}\sum_2^\infty I_n \sin((n+1)wt+\vartheta_n) + \sqrt{2}\sum_2^\infty I_n \sin((n-1)wt+\vartheta_n) \end{aligned} \tag{6}$$

Therefore, so long as detect active current i_{1p} (t) and reactive current i_{1q} (t), then, harmonic current $i_h(t)$ can be expressed as $i_h(t) = i_L(t) - i_{1p}(t) - i_{1q}(t)$, if APF need to compensate harmonics and reactive power simultaneously, cut the channel of i_{1q} in the detecting circuit that shown in Fig.3.

3.2 Obtain Sine and Cosine Signal That in Phase of The Power Supply Voltage

The i_p-i_q detecting method for harmonics that based on instantaneous reactive power theory, which need a PLL to obtain sine signal sinwt and cosine signal coswt that in phase of power voltage u_s [2,], [4]. However, in practical application, PLL is easily disturbed by extra signal, there will certain be some delay for obtaining sine and cosine signal, meanwhile, increases PLL and signal generate circuit will increase the difficulty of design and debug of the circuit, so, there will be some influence to the detecting results. In this paper, we replace PLL and signal generated circuit by dashed line frame 2 that shown in Fig.2 that use the relation of differential. First, let power supply voltage U_s pass a LPF to avoid the impact of voltage distortion, which Will be discussed in detail below. Then, divide power supply voltage U_s by its peak value, which can get sine signal sinwt, afterwards differential sine signal and then divided by 314 (because of angular frequency $w = 2\pi f$), then we can easily obtained the same frequency cosine signal coswt.

This method replace PLL and signal generate circuit by a simple circuit, which make the detection circuit more concise and avoid the disturbance of extra signal.

3.3 Impact on Detecting Results when Voltage Distortion

In order to avoid the impact of voltage distortion, we introduce the part of dashed line frame 1 that shown in Fig.2, let power supply voltage pass a LPF, whose function is to separate the fundamental wave component from power supply voltage. We will get the sine signal $\sin(wt+\theta)$ and cosine signal $\cos(wt+\theta)$ through the circuit of dashed line frame 2, where, θ is the phase shift when fundamental wave component of voltage pass LPF and the circuit of dashed line frame 2. The same analysis as chapter 2.1, multiply both sides of Eqs.(1) by $2\sin(wt+\theta)$ and $2\cos(wt+\theta)$, then, separate the DC component through LPF respectively, and then multiply the results by $2\sin(wt+\theta)$ and $2\cos(wt+\theta)$ respectively, finally, add the results together, which is $I_p$sinwt+$I_q$coswt. It can be seen that the detecting results is not affected by the

distortion of power supply voltage since introducing the circuit of dashed line frame 1 that shown in Fig.2. Limited by the length, the inferential process above is not given in detail, only give the analysis result.

3.4 Propose PID Negative Feedback

Considering the inherent delay of LPF in the detection, the results will be subject to some certain influences. Meanwhile, in order to improve the dynamic performance when loads change, the feedback of harmonics and reactive currents is proposed[14], [15].The negative feedback of PID is presented in this paper, which will compensate the error that caused by time delay. This method will supplement DC component of instantaneous power for power supply voltage and current, meanwhile, it will provide forecast signal in advance to power supply current, which will enhance the dynamic response speed of detection circuit. When in stable state, the DC component of instantaneous power can be filter by LPF, so,the detection result is not affected by PID negative feedback. The parameters of PID can be regulated step-by-step by Matlab simulation.

4 Hysteresis Current Controller Based on Neural Network

The hysteresis current controller with a fixed band derives the switching signals from the inverter according to the comparison of the current error to keep the current within the hysteresis band. When the current hysteresis controller operated, the system storage devices do not need specific parameters, therefore it can be replaced by BP neural network. It will achieve hysteresis control functions by training a neural network controller on the basis of hysteresis control ideas.

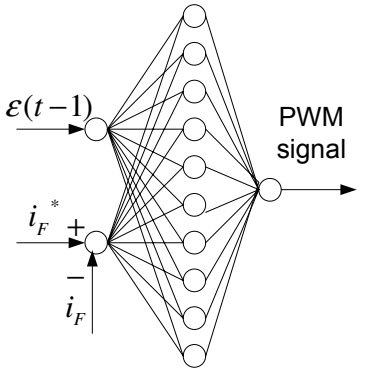


Fig. 3. The diagram of the NN hysteresis controller

Fig.3 shows the diagram of NN hysteresis controller**.** After the APF compensation actual current being compareded with the detected reference current, the error signal

was sent into the neural network. After computing, the outputted PWM switching signal will drive the switching of APF main circuit [16].

BP (back-propagation) network is a multi-network for the weight value of nonlinear differentiable function. In this paper, we adopt an improved BP algorithm, which is the L-M (levenberg-marquardt) algorithm. L-M is an algorithm based on numerical optimization, which combines the advantages of the gradient method and Newton's Law. It has a rapid convergence speed and excellent performance. In this L-M, it dopes a three-layer BP neural network which contains a hidden layer that is 2-10-1 structure, as shown in Fig.3. We take two neurons from the input layer, corresponding time t and t-1 moment of the current error signal $\varepsilon(t)$ and $\varepsilon(t-1)$. One neuron in output layer is corresponding to APF switching control signals. Hidden and output layer apply S-type activate function.

$\varepsilon_i = i_F^* - i_F$ is ANN (artificial neural network) input after being cooperated, where i_F is the actual current of APF compensation and i_F^* is instruction current that detected by detection circuit.

ANN comparator output is proposed:

$$u = f_2\left\{\sum_{j=1}^{L} w_{2jk} f_1\left[\sum_{i=1}^{n} w_{1ij}\varepsilon_i\right]\right\} \tag{7}$$

Where, f_1 is the S-type log-sigmoid activation function of the hidden layer.

$$f_{1pj}^r = 1/(1+\exp(-a_{1pj}^r)) \tag{8}$$

Where, f_{1pj}^r is the output at the time of the r step iteration when the jth neuron node corresponds sample input of the pth group of the hidden layer and a_{1pj}^r is the input weighting sum total of the input layer which is received at the time of the r step iteration when the jth neuron node corresponds sample input of the pth group of the hidden layer.

$$a_{1pj}^r = \sum_{i=1}^{n} w_{1ij}\varepsilon_i \tag{9}$$

Where, f_2 is the S-type log-sigmoid activation function of the output layer.

$$f_{2jk}^r = 1/(1+\exp(-a_{2jk}^r)) \tag{10}$$

Where, f_{2jk}^r is the kth neuron node's output of the output layer at the time of the r step iteration. The hidden layer's output is as output layer's input.

$$a_{2jk}^r = \sum_{j=1}^{L} w_{2jk} f_{1pj}^r \tag{9}$$

The error signal is the squares' sum of output error as follows:

$$E = \frac{1}{2}\sum_{k=1}^{M}(t_k - u_k)^2 \tag{9}$$

Where, t_k is the output expected value of the kth node and u_k is the output actual value of the kth node. After the training, the optimal values are approached to the weight vector of neural network and the neural network' output approaches the desired PWM waveform of hysteresis controller.

Then the APF reference currents i_F^* compared with APF feedback currents i_F and the error signals are operated by the NN hysteresis current controller to generate the firing pulses which activate the inverter power switches in a manner that reduces the current error. Let h_b be the width of the hysteresis band around respective phase APF currents, then the equation for the fixed band is as follows:

$$i_F^* = i_{\max} \sin(\omega t) \tag{13}$$

$$i_{up} = i_F^* + h_b \tag{14}$$

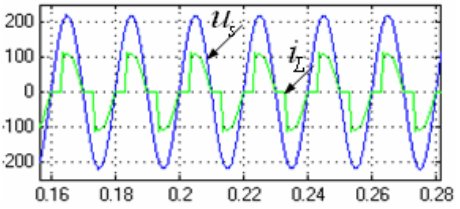
$$i_{low} = i_F^* - h_b \tag{15}$$

Where, i_{up} is the upper band and i_{low} is the lower band. If the APF current feedback $i_F > (i_F^* + h_b)$, and G_1 =-1, which means that the inverter output voltage is negative, in order to keep the current within the hysteresis band. If the APF current feedback $i_F < (i_F^* - h_b)$, G_1 = 1, which means that the inverter output voltage switches to positive, in order to increase the actual current..

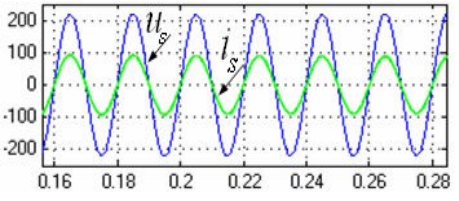
5 Simulation Study

In order to confirm the accuracy and validity of the proposed method, this paper build simulation model and analyse with Matlab 6.5. The system constitution of single-phase SAPF is shown in Fig.1, according to this figure and Fig.2, Fig.3, select the corresponding subsystem module and constitute the specific simulation circuit, and replace nonlinear load by single-phase resistance load rectifier. The simulation parameter is: input voltage u_s=220v, rectifier bridge load, R= 2Ω, APF inductance L_0=0.01H, APF DC link capacitor C=200UF, APF DC link voltage Vc=450V. Fig.2 shows the harmonics detection circuit, LPF of dashed line frame 1 is selected for the type of Butterworth, whose order number is two and cutoff frequency is 60 HZ, whereas, other two LPF are selected for the type of Butterworth, whose order number is two and cutoff frequency is 25HZ.

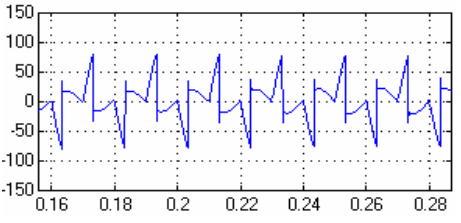
Simulation results are shown in Fig.4 and Fig.5. Fig.4 shows the simulation results with SAPF, from source voltage, source current without SAPF and source current with SAPF, we can see that source current i_L exist distortion, whereas, when used SAPF, both source voltage and current i_s are sinusoidal, meanwhile, source current is completely in phase with source voltage. So, the SAPF can Suppress harmonics and compensate reactive power simultaneously. Fig.4(c) shows the waveform of compensation current. Fig.5 shows the spectrum of source current without and with SAPF, we can see that the Total Harmonic Distortion (THD) drop from 36.56% to 0.19% with SAPF. Therefore, this SAPF has a good performance in harmonics suppress.



(a) Source voltage and current without SAPF

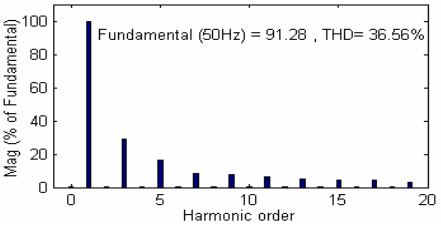


(b) Source voltage and current with SAPF

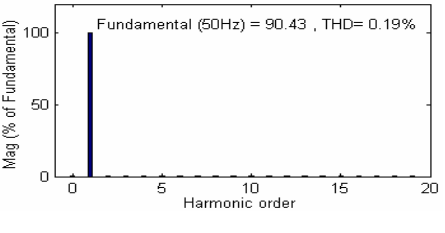


(c) Compensation current

Fig. 4. Simulation results with SAPF



(a) Spectrum of source current without SAPF



(b) Spectrum of source current with SAPF

Fig. 5. Spectrum analysis

6 Conclusion

The filter effect of SAPF is decided by the accuracy and real-time of power source harmonic currents detection and the speed and accuracy of current tracking control. This paper presents a new detecting method of harmonics and reactive currents for single-phase circuit, which is simple, time delay is short and have a good performance in accuracy and real-time. Meanwhile, this paper also presents a new hysteresis current controller with ANN, which makes actual output current of SAPF can accurately track the change of instruction current. The simulation results indicate that this method can effectively compensate reactive power and suppress harmonics. If apply this method to single-phase circuit (such as electrified railway) for harmonics suppression, which can greatly improve the quality of power supply and good for highly effective and steady operation for power grids.

References

[1] Arrilliga, J., Bradley, D.A.: Power System Harmonics. Huazhong University of Science and Technology Press, Wuhan (1994)
[2] Wang, Z.-a., Yang, J., Liu, J.-j.: Harmonics Suppress and Reactive Power Compensation. Mechanical Industry Press, Beijing (1998)
[3] Ren, Y.-f., Li, H.-s., He, G., et al.: The Analysis of Two Real-Time Detecting Approaches of Instantaneous Harmonic and Reactive Current in Single-phase Circuit. Proceeding of the CSUEPSA 15(1), 95–98 (2003)
[4] Akagi, H., Kanazawa, Y., Nabae, A.: Generalized Therory of the Instantaneous Reactive Power in Three-phase Circuit. In: IEEE&JIEE Proceeding IPEC, Tokyo, pp. 1375–1386 (1983)
[5] Sun, S.-g., Wang, J.-q., Shi, S.-q.: Syudy on Real-time Dection of Harmonics and Reactive Currents in Single phase Circuit. Electrical Measurement & Instrumentation 505(45), 4–7 (2008)
[6] Fan, X.-b., Zhan, D.-r., Sun, Q., et al.: Hystereis Current Control Strategy for Three-phase Three-wire Active Power Filter. Automation of Electric Power Systems 31(18), 57–60 (2007)
[7] Lu, X.-l., Zhang, S.-h., Cao, C.-k., et al.: New Detecting Method for Harmonics and Reactive Currents in Single-phase Circuits. High Voltage Engineering 33(3), 163–166 (2007)
[8] Rong, F., Luo, A., Fan, Q.: Novel Harmonics Current Detecting Algorithm. High Voltage Engineering 34(1), 138–141 (2008)
[9] Bhattacharaya, S., Divan, D.: Synchronous Flame Based Controller Implementation for a Hybrid Series Active Filter System. In: Proceeding of 1995 IEEE/IAS Annual Meeting, Orlando, USA, pp. 2531–2540 (1995)
[10] Wang, L., Liu, H.-J., Wang, C.: Summary of the Instantaneous Reactive Power Theory. High Voltage Engineering 32(2), 98–101 (2006)
[11] Li, S.-q., Peng, Y.-l., Zhou, Y.-q.: Study on An Improved Adaptive Harmonic Current Detecting Method. High Voltage Engineering 28(12), 3–5 (2002)
[12] Yang, J., Wang, Z.-a., Qiu, G.-y.: A Detecting Method for Harmonics and Reactive Currents in Single-phase Circuit. Trans. of China Electrotechnical Society 11(3), 42–46 (1996)
[13] Hu, M., Chen, H.: Active Power Filter Technology and Its Application. Automation of Electric Power Systems 2(3), 66–69 (2002)

[14] Tao, J., Liu, Z.-z.: Syudy on the Detecting Approach of Harmonics and Reactive Currents. Automation of Electric Power Systems 31(1), 31–33 (2001)
[15] Sun, S.-h., Li, P.: A Directly Detection Method for Harmonics and Reactive Currents. Automation of Electric Power Systems 26(19), 52–55 (2002)
[16] Harashima, F., Demizu, Y., Kondo, S., et al.: Application of neural networks to power converter control. In: Conf. Rec. IEEE2IAS Annu. Meeting, Conf. Rec., pp. 1087–1091 (1989)

Qualitative Spatio-temporal Reasoning about Moving Objects in Three-Dimensional Space

Jingde Cheng

Department of Information and Computer Sciences
Saitama University, Saitama, 338-8570, Japan
cheng@ics.saitama-u.ac.jp

Abstract. To represent and reason about moving objects in three-dimensional space qualitatively, we need a right fundamental logic system to provide us with a criterion of logical validity for reasoning as well as a formal representation language. In order to reason about new spatio-temporal knowledge with incomplete or sometime even inconsistent knowledge, the fundamental logic must be able to underlie relevant and truth-preserving reasoning in the sense of conditional, ampliative reasoning, paracomplete and paraconsistent reasoning, and spatio-temporal reasoning. This paper proposes a new family of three-dimensional spatio-temporal relevant logics as a hopeful candidate for the fundamental logic, and shows that the logics can satisfy all the requirements for the fundamental logic we need.

Keywords: Relevant and truth-preserving reasoning, Ampliative reasoning, Paracomplete and paraconsistent reasoning, Spatio-temporal reasoning, Three-dimensional spatio-temporal relevant logic.

1 Introduction

Time and space are two essential and primitive parts of the fundamental structure of the universe as well as the fundamental intellectual structure of our human beings. Temporal, spatial, and spatio-temporal notions, i.e., time, schedule, time zone, shape, size, distance, orientation, relative position, connectivity, motion, speed, etc, play many important roles in our cognition and understanding of the real world and our communication and cooperation in the human society. Our temporal, spatial, and spatio-temporal knowledge are fundamentals and sources from which we can represent, reason about, and derive new knowledge. There are many applications that need means for representing and reasoning about temporal, spatial, and spatio-temporal knowledge, such as robotics, motion planning, motion capture, machine vision, solid modeling, natural language understanding, temporal database systems, spatial database systems, geographic information systems, distributed systems, air traffic control systems, train control systems, expressway control systems, etc. Therefore, it is certainly a very important task to provide the means dealing with temporal, spatial, and spatio-temporal knowledge required by various applications.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 637–648, 2008.
© Springer-Verlag Berlin Heidelberg 2008

To develop and maintain any intelligent system dealing with moving objects in three-dimensional space, we have to consider not only plane (two-dimensional) space but also solid (three-dimensional) space. To represent and reason about moving objects in three-dimensional space, we need a right fundamental logic system to provide us with a criterion of logical validity for reasoning as well as a formal representation language. In order to reason about new spatio-temporal knowledge with incomplete or sometime even inconsistent knowledge, the fundamental logic must be able to underlie relevant and truth-preserving reasoning in the sense of conditional, ampliative reasoning, paracomplete and paraconsistent reasoning, three-dimensional spatial reasoning, temporal reasoning, and three-dimensional spatio-temporal reasoning.

There are many classical or modal logic systems proposed for representing and reasoning about temporal, spatial, and spatio-temporal knowledge [3, 11-14, 16, 17, 19-21]. All of these logics are some how based on classical mathematical logic, and therefore, they cannot underlie relevant and truth-preserving reasoning in the sense of conditional, ampliative reasoning, paracomplete and paraconsistent reasoning. Relevant logics cannot underlie spatial reasoning, temporal reasoning, and spatio-temporal reasoning, even though them can underlie relevant and truth-preserving reasoning in the sense of conditional, ampliative reasoning, paracomplete and paraconsistent reasoning well [1, 2, 15, 18]. Two-dimensional spatio-temporal relevant logics cannot underlie three-dimensional spatial reasoning and three-dimensional spatio-temporal reasoning [8]. Therefore, no existing logic system can satisfy all of the above requirements. This paper proposes a new family of three-dimensional spatio-temporal relevant logics as a hopeful candidate for the fundamental logic, and shows that the logics can satisfy all the requirements for the fundamental logic we need.

2 The Logical Basis for Qualitative Spatio-temporal Reasoning about Moving Objects in Three-Dimensional Space

The question, "Which is the right logic?" invites the immediate counter-question "Right for what?" Only if we certainly know what we need, we can make a good choice. The present author considers that the fundamental logic system to underlie representing and reasoning about moving objects in three-dimensional space must satisfy all the following essential requirements.

First, as a general logical criterion for the validity of reasoning, the logic must be able to underlie relevant and truth-preserving reasoning in the sense of conditional, i.e., for any reasoning based on the logic to be valid, if its premises are true in the sense of conditional, then its conclusion must be relevant to the premises and true in the sense of conditional.

Second, the logic must be able to underlie ampliative reasoning in the sense that the truth of conclusion of the reasoning should be recognized after the completion of the reasoning process but not be invoked in deciding the truth of premises of the reasoning. From the viewpoint to regard reasoning as the process of drawing new conclusions from given premises, any meaningful reasoning must be ampliative but not circular and/or tautological.

Third, the logic must be able to underlie paracomplete and paraconsistent reasoning. In particular, the so-called principle of Explosion that everything follows from a

contradiction should not be accepted by the logic as a valid principle. In general, our knowledge about a domain as well as a scientific discipline may be incomplete and/or inconsistent in many ways, i.e., it gives us no evidence for deciding the truth of either a proposition or its negation, and/or it directly or indirectly includes some contradictions. Therefore, reasoning with incomplete and/or inconsistent knowledge is the rule rather than the exception in our everyday lives and almost all scientific disciplines.

Finally, the logic must be able to underlie temporal, three-dimensional spatial, and three-dimensional spatio-temporal reasoning. In an application system dealing with mobile objects in three-dimensional space, any reasoning may somehow depend on notions of time and/or three-dimensional space. Not only propositions (statements) about mobile objects but also relevant relationships among mobile objects may be dependent on three-dimensional spatial regions and/or points, and may change over time. Some properties of mobile objects, i.e., motion and speed, are intrinsically depend on both time and three-dimensional space. This naturally requires that the logic must be able to underlie temporal, three-dimensional spatial, and three-dimensional spatio-temporal reasoning.

Classical mathematical logic (**CML** for short) was established based on a number of fundamental assumptions and/or principles. Among them, the most characteristic one is the classical account of validity (i.e., an argument/reasoning is valid if and only if it is impossible for all its premises to be true while its conclusion is false) that is the logical validity criterion of **CML** by which one can decide whether the conclusion of an argument/reasoning really does follow from its premises or not in the framework of **CML**. Because relevance between premises and conclusion of an argument/reasoning is not accounted for by the classical validity criterion, a reasoning based on **CML** is not necessarily relevant. On the other hand, in **CML** the notion of conditional, which is intrinsically intensional but not truth-functional, is represented by the notion of material implication, which is intrinsically an extensional truth-function. This leads to the problem of 'implicational paradoxes' [1, 2, 15, 18] (therefore any reasoning based on **CML** is not truth-preserving in the sense of conditional) as well as the problem that a reasoning based on **CML** must be circular and/or tautological but not ampliative. Moreover, because **CML** accepts the principle of Explosion (i.e., everything follows from a contradiction), reasoning under inconsistency is impossible within the framework of **CML**. The above three facts are also true to those classical conservative extensions or non-classical alternatives of **CML** where the classical account of validity is adopted as the logical validity criterion and the notion of conditional is directly or indirectly represented by the material implication. Finally **CML** does not provide explicit means to deal with the notion of time and/or the notion of space. Therefore, **CML** cannot satisfy any of the essential requirements for the fundamental logic system.

Temporal (classical) logics was established in order to represent and reason about notions, relations, and properties of time-related entities within a logical framework, and therefore to underlie temporal reasoning, i.e., reasoning about those propositions and/or formulas whose truth-values may depend on time [4, 22]. However, because any temporal (classical) logic is a classical conservative extension of **CML** in the sense that it is based on the classical account of validity and it represents the notion of conditional directly or indirectly by the material implication, all problems in **CML** caused by the classical account of validity and the material implication also remain in

temporal (classical) logic. As a result, no temporal (classical) logic can satisfy the first three of the essential requirements for the fundamental logic system.

Spatial (classical) logics was proposed in order to deal with geometric and/or topological entities, notions, relations, and properties, and therefore to underlie spatial reasoning, i.e., reasoning about those propositions and formulas whose truth-values may depend on a location [11-14, 20]. However, these existing spatial logics are classical conservative extensions of **CML** in the sense that they are based on the classical account of validity and they represent the notion of conditional directly or indirectly by the material implication. Therefore, similar to the case of temporal (classical) logic, these spatial logics cannot satisfy the first three of the essential requirements for the fundamental logic system.

Traditional relevant (relevance) logics were constructed during the 1950s in order to find a mathematically satisfactory way of grasping the elusive notion of relevance of antecedent to consequent in conditionals, and to obtain a notion of implication which is free from the so-called 'paradoxes' of material and strict implication [1, 2, 15, 18]. Some major traditional relevant logic systems are 'system **E** of entailment', 'system **R** of relevant implication', and 'system **T** of ticket entailment'. A major characteristic of the relevant logics is that they have a primitive intensional connective to represent the notion of conditional and their logical theorems include no implicational paradoxes. The underlying principle of the relevant logics is the relevance principle, i.e., for any entailment provable in **E**, **R**, or **T**, its antecedent and consequent must share a propositional variable. Variable-sharing is a formal notion designed to reflect the idea that there be a meaning-connection between the antecedent and consequent of an entailment. It is this relevance principle that excludes those implicational paradoxes from logical axioms or theorems of relevant logics. In order to establish a satisfactory logic calculus of conditional to underlie relevant reasoning, the present author has proposed some strong relevant (relevance) logics, named **Rc**, **Ec**, and **Tc** [5, 6, 9]. The logics require that the premises of an argument represented by a conditional include no unnecessary and needless conjuncts and the conclusion of that argument includes no unnecessary and needless disjuncts. As a modification of traditional relevant logics **R**, **E**, and **T**, strong relevant logics **Rc**, **Ec**, and **Tc** rejects all conjunction-implicational paradoxes and disjunction-implicational paradoxes in **R**, **E**, and **T**, respectively. What underlies the strong relevant logics is the strong relevance principle: If A is a theorem of **Rc**, **Ec**, or **Tc**, then every propositional variable in A occurs at least once as an antecedent part and at least once as a consequent part. Although strong relevant logics can satisfy the first three of the essential requirements for the fundamental logic system, they do not provide explicit means to deal with the notion of time and/or the notion of space and therefore they cannot satisfy the fourth essential requirement.

Two-dimensional spatio-temporal relevant logics [8], which are obtained by introducing region connection predicates and axiom schemata of spatial logic RCC [3, 11-14], point position predicates and axiom schemata, and point adjacency predicates and axiom schemata into temporal relevant logics [7], can satisfy the first three of the essential requirements for the fundamental logic system but only can partly satisfy the fourth essential requirement because they cannot deal with spatio-temporal notions (such as motion and speed) and three-dimensional space.

Therefore, no existing logic system can satisfy all of the essential requirements.

3 Three-Dimensional Spatio-temporal Relevant Logic

We now propose a new family of relevant logic systems, named ***three-dimensional spatio-temporal relevant logic***, which can satisfy all the essential requirements for the fundamental logic system to underlie representing and reasoning about mobile three-dimensional geometric objects. The logics are obtained by introducing predicates and axiom schemata about solid-region connection, predicates and axiom schemata about point position, and predicates and axiom schemata about motion of mobile objects into temporal relevant logics [7]. Therefore, they are conservative extensions of temporal relevant logics as well as strong relevant logics. On the other hand, we do not introduce primitive predicates about the notion of distance but define the notion of distance by predicates about point position and adjacency, predicates about movement of mobile objects, and temporal operators.

Let $\{r_1, r_2, r_3, \ldots\}$ be a countably infinite set of individual variables, called ***solid-region variables***. Atomic formulas of the form $\boldsymbol{C}(r_1, r_2)$ are read as 'region r_1 connects with region r_2.' Let $\{p_1, p_2, p_3, \ldots\}$ be a countably infinite set of individual variables, called ***point variables***. Let ***TCP*** be an individual constant of point, called ***the central point***. Atomic formulas of the form $\boldsymbol{I}(p_1, r_1)$ are read as 'point p_1 is included in region r_1.' Atomic formulas of the form $\boldsymbol{Id}(p_1, p_2)$ are read as 'point p_1 is identical with p_2.' Atomic formulas of the form $\boldsymbol{Arc}(p_1, p_2)$ are read as 'points p_1, p_2 are adjacent such that there is an arc from point p_1 to point p_2, or more simply, point p_1 is adjacent to point p_2.' Note that an arc has a direction. Atomic formulas of the form $\boldsymbol{Reachable}(p_1, p_2)$ are read as 'there is at least one directed path (i.e., a sequence of arcs such that one connects to the next one) from point p_1 to point p_2.' Atomic formulas of the form $\boldsymbol{NH}(p_1, p_2)$ are read as 'taking ***TCP*** as the reference point, the position of point p_1 is not higher than the position of point p_2, i.e., the length of the vertical line from p_1 to ***TCP*** is not longer than the length of the vertical line from p_2 to ***TCP***. Here, $\boldsymbol{C}(r_1, r_2)$, $\boldsymbol{I}(p_1, r_1)$, $\boldsymbol{Id}(p_1, p_2)$, $\boldsymbol{Arc}(p_1, p_2)$, $\boldsymbol{Reachable}(p_1, p_2)$ and $\boldsymbol{NH}(p_1, p_2)$ are primitive binary predicates to represent three-dimensional geometric relationships between three-dimensional geometric regions and points. Note that here we use a many-sorted language.

Let $\{o_1, o_2, o_3, \ldots\}$ be a countably infinite set of individual variables, called ***object variables***. Atomic formulas of the form $\boldsymbol{A}(o_1, p_1)$ are read as 'object o_1 arrives at point p_1.' Atomic formulas of the form $\boldsymbol{NS}(o_1, o_2)$ are read as 'the speed of object o_1 is not faster than the speed of object o_2. Here, $\boldsymbol{A}(o_1, p_1)$ and $\boldsymbol{NS}(o_1, o_2)$ are primitive binary predicates to represent motion relationships between mobile objects in a three-dimensional geometric space.

The symbols (logical connectives, quantifiers, individual variables, individual constants, solid-region variables, object variables, predicates, temporal operators), region connection predicates, point position predicates, object movement predicates, axiom schemata, and inference rules are as follows:

Symbols

$\{\neg, \Rightarrow, \wedge, \forall, \exists, x_1, x_2, ..., x_n, ..., c_1, c_2, ..., c_n, ..., r_1, r_2, r_3, \ldots, r_n, \ldots, o_1, o_2, o_3, \ldots, o_n, \ldots, p_0^1, ..., p_0^n, ..., p_1^1, ..., p_1^n, ..., p_2^1, ..., p_2^n, ..., p_k^1, ..., p_k^n, ..., (,), \boldsymbol{G}, \boldsymbol{H}, \boldsymbol{F}, \boldsymbol{P}\}$

Primitive logical connectives

$\Rightarrow$ (conjunction, $A \otimes B =_{df} \neg(A \Rightarrow \neg B)$),
$\oplus$ (intensional disjunction, $A \oplus B =_{df} \neg A \Rightarrow B$),
$\Leftrightarrow$ (intensional equivalence, $A \Leftrightarrow B =_{df} (A \Rightarrow B) \otimes (B \Rightarrow A)$),
$\vee$ (extensional disjunction, $A \vee B =_{df} \neg(\neg A \wedge \neg B)$),
$\rightarrow$ (material implication, $A \rightarrow B =_{df} \neg(A \wedge \neg B)$ or $\neg A \vee B$),
$\leftrightarrow$ (extensional equivalence, $A \leftrightarrow B =_{df} (A \rightarrow B) \wedge (B \rightarrow A)$).

Temporal operators

G (future-tense always or henceforth operator, ***GA*** means 'it will always be the case in the future from now that *A*'),
H (past-tense always operator, ***HA*** means 'it has always been the case in the past up to now that *A*'),
F (future-tense sometime or eventually operator, ***FA*** means 'it will be the case at least once in the future from now that *A*'),
P (past-tense sometime operator, ***PA*** means 'it has been the case at least once in the past up to now that *A*').

Note that these temporal operators are not independent and can be defined as follows: $\boldsymbol{GA} =_{df} \neg \boldsymbol{F} \neg A$, $\boldsymbol{HA} =_{df} \neg \boldsymbol{P} \neg A$, $\boldsymbol{FA} =_{df} \neg \boldsymbol{G} \neg A$, $\boldsymbol{PA} =_{df} \neg \boldsymbol{H} \neg A$.

Primitive binary predicates

$\boldsymbol{C}(r_1, r_2)$ (connection), $\boldsymbol{I}(p_1, r_1)$ (inclusion), $\boldsymbol{Id}(p_1, p_2)$ (the same point), $\boldsymbol{Arc}(p_1, p_2)$ (arc), $\boldsymbol{Reachable}(p_1, p_2)$ (reachable), $\boldsymbol{NH}(p_1, p_2)$ (not higher than), $\boldsymbol{NS}(o_1, o_2)$ (not speedier than).

Defined binary predicates

$\boldsymbol{DC}(r_1, r_2) =_{df} \neg \boldsymbol{C}(r_1, r_2)$, $\boldsymbol{DC}(r_1, r_2) =_{df} \neg(\exists p_1(\boldsymbol{I}(p_1, r_1) \wedge \boldsymbol{I}(p_1, r_2)))$ ($\boldsymbol{DC}(r_1, r_2)$ means 'r_1 is disconnected from r_2')
$\boldsymbol{Pa}(r_1, r_2) =_{df} \forall r_3(\boldsymbol{C}(r_3, r_1) \Rightarrow \boldsymbol{C}(r_3, r_2))$, $\boldsymbol{Pa}(r_1, r_2) =_{df} \forall p_1(\boldsymbol{I}(p_1, r_1) \Rightarrow \boldsymbol{I}(p_1, r_2))$ ($\boldsymbol{Pa}(r_1, r_2)$ means 'r_1 is a part of r_2')
$\boldsymbol{PrPa}(r_1, r_2) =_{df} \boldsymbol{Pa}(r_1, r_2) \wedge (\neg \boldsymbol{Pa}(r_2, r_1))$ ($\boldsymbol{PrPa}(r_1, r_2)$ means 'r_1 is a proper part of r_2')
$\boldsymbol{EQ}(r_1, r_2) =_{df} \boldsymbol{Pa}(r_1, r_2) \wedge \boldsymbol{Pa}(r_2, r_1)$ ($\boldsymbol{EQ}(r_1, r_2)$ means 'r_1 is identical with r_2')
$\boldsymbol{O}(r_1, r_2) =_{df} \exists r_3(\boldsymbol{Pa}(r_3, r_1) \wedge \boldsymbol{Pa}(r_3, r_2))$ ($\boldsymbol{O}(r_1, r_2)$ means 'r_1 overlaps r_2')
$\boldsymbol{DR}(r_1, r_2) =_{df} \neg \boldsymbol{O}(r_1, r_2)$ ($\boldsymbol{DR}(r_1, r_2)$ means 'r_1 is discrete from r_2')
$\boldsymbol{PaO}(r_1, r_2) =_{df} \boldsymbol{O}(r_1, r_2) \wedge (\neg \boldsymbol{Pa}(r_1, r_2)) \wedge (\neg \boldsymbol{Pa}(r_2, r_1))$ ($\boldsymbol{PaO}(r_1, r_2)$ means 'r_1 partially overlaps r_2')
$\boldsymbol{EC}(r_1, r_2) =_{df} \boldsymbol{C}(r_1, r_2) \wedge (\neg \boldsymbol{O}(r_1, r_2))$ ($\boldsymbol{EC}(r_1, r_2)$ means 'r_1 is externally connected to r_2')
$\boldsymbol{TPrPa}(r_1, r_2) =_{df} \boldsymbol{PrPa}(r_1, r_2) \wedge \exists r_3(\boldsymbol{EC}(r_3, r_1) \wedge \boldsymbol{EC}(r_3, r_2))$ ($\boldsymbol{TPrPa}(r_1, r_2)$ means 'r_1 is a tangential proper part of r_2')
$\boldsymbol{NTPrPa}(r_1, r_2) =_{df} \boldsymbol{PrPa}(r_1, r_2) \wedge (\neg \exists r_3(\boldsymbol{EC}(r_3, r_1) \wedge \boldsymbol{EC}(r_3, r_2)))$ ($\boldsymbol{NTPrPa}(r_1, r_2)$ means 'r_1 is a nontangential proper part of r_2')
$\boldsymbol{SA}(p_1, p_2) =_{df} \boldsymbol{NH}(p_1, p_2) \wedge \boldsymbol{NH}(p_2, p_1)$ ($\boldsymbol{SA}(p_1, p_2)$ means 'the position of point p_1 and the position of point p_2 are in the same altitude')
$\boldsymbol{Hi}(p_1, p_2) =_{df} \neg \boldsymbol{NH}(p_1, p_2)$ ($\boldsymbol{Hi}(p_1, p_2)$ means 'the position of point p_1 is higher than the position of point p_2')

$\boldsymbol{SS}(o_1, o_2) =_{df} \boldsymbol{NS}(o_1, o_2) \wedge \boldsymbol{NS}(o_2, o_1)$ ($\boldsymbol{SS}(o_1, o_2)$ means 'the motion of object o_1 and the motion of object o_2 are in the same speed')

$\boldsymbol{Sp}(o_1, o_2) =_{df} \neg\boldsymbol{NS}(o_1, o_2)$ ($\boldsymbol{Sp}(o_1, o_2)$ means 'the p motion of object o_1 is faster than the motion of object o_2')

$\boldsymbol{ND}(p_1, p_2, p_3) =_{df} \exists o_1 \exists o_2((\boldsymbol{A}(o_1, p_1) \wedge \boldsymbol{A}(o_2, p_2) \wedge \boldsymbol{Reachable}(p_1, p_3) \wedge \boldsymbol{Reachable}(p_2, p_3) \wedge \boldsymbol{SS}(o_1, o_2)) \Rightarrow \boldsymbol{G}(\boldsymbol{A}(o_1, p_3) \Rightarrow \boldsymbol{A}(o_2, p_3)))$ ($\boldsymbol{ND}(p_1, p_2, p_3)$ means 'the distance of between point p_2 and point p_3 is not more distant than the distance of between point p_1 and point p_3')

$\boldsymbol{SD}(p_1, p_2, p_3) =_{df} \boldsymbol{ND}(p_1, p_2, p_3) \wedge \boldsymbol{ND}(p_2, p_1, p_3)$ ($\boldsymbol{SD}(p_1, p_2, p_3)$ means 'the distance of between point p_1 and point p_3 is equal to the distance of between point p_2 and point p_3')

$\boldsymbol{Ne}(p_1, p_2, p_3) =_{df} \neg\boldsymbol{ND}(p_1, p_2, p_3)$ ($\boldsymbol{Ne}(p_1, p_2, p_3)$ means 'the distance of between point p_1 and point p_3 is nearer than the distance of between point p_2 and point p_3')

Axiom schemata

E1: $A \Rightarrow A$,

E2: $(A \Rightarrow B) \Rightarrow ((C \Rightarrow A) \Rightarrow (C \Rightarrow B))$,

E2′: $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))$,

E3: $(A \Rightarrow (A \Rightarrow B)) \Rightarrow (A \Rightarrow B)$,

E3′: $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$,

E3″: $(A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C))$,

E4: $(A \Rightarrow ((B \Rightarrow C) \Rightarrow D)) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow D))$,

E4′: $(A \Rightarrow B) \Rightarrow (((A \Rightarrow B) \Rightarrow C) \Rightarrow C)$,

E4″: $((A \Rightarrow A) \Rightarrow B) \Rightarrow B$,

E4‴: $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (((A \Rightarrow C) \Rightarrow D) \Rightarrow D))$,

E5: $(A \Rightarrow (B \Rightarrow C)) \Rightarrow (B \Rightarrow (A \Rightarrow C))$,

E5′: $A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$,

N1: $(A \Rightarrow (\neg A)) \Rightarrow (\neg A)$, N2: $(A \Rightarrow (\neg B)) \Rightarrow (B \Rightarrow (\neg A))$, N3: $(\neg(\neg A)) \Rightarrow A$,

C1: $(A \wedge B) \Rightarrow A$, C2: $(A \wedge B) \Rightarrow B$, C3: $((A \Rightarrow B) \wedge (A \Rightarrow C)) \Rightarrow (A \Rightarrow (B \wedge C))$,

C4: $(\boldsymbol{L}A \wedge \boldsymbol{L}B) \Rightarrow \boldsymbol{L}(A \wedge B)$, where $\boldsymbol{L}A =_{df} (A \Rightarrow A) \Rightarrow A$,

D1: $A \Rightarrow (A \vee B)$, D2: $B \Rightarrow (A \vee B)$, D3: $((A \Rightarrow C) \wedge (B \Rightarrow C)) \Rightarrow ((A \vee B) \Rightarrow C)$,

DCD: $(A \wedge (B \vee C)) \Rightarrow ((A \wedge B) \vee C)$,

C5: $(A \wedge A) \Rightarrow A$, C6: $(A \wedge B) \Rightarrow (B \wedge A)$, C7: $((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$,

C8: $(A \wedge (A \Rightarrow B)) \Rightarrow B$, C9: $\neg(A \wedge \neg A)$, C10: $A \Rightarrow (B \Rightarrow (A \wedge B))$,

T1: $\boldsymbol{G}(A \Rightarrow B) \Rightarrow (\boldsymbol{G}A \Rightarrow \boldsymbol{G}B)$, T2: $\boldsymbol{H}(A \Rightarrow B) \Rightarrow (\boldsymbol{H}A \Rightarrow \boldsymbol{H}B)$,

T3: $A \Rightarrow \boldsymbol{G}(\boldsymbol{P}A)$, T4: $A \Rightarrow \boldsymbol{H}(\boldsymbol{F}A)$, T5: $\boldsymbol{G}A \Rightarrow \boldsymbol{G}(\boldsymbol{G}A)$,

T6: $(\boldsymbol{F}A \wedge \boldsymbol{F}B) \Rightarrow (\boldsymbol{F}(A \wedge \boldsymbol{F}B) \vee \boldsymbol{F}(A \wedge B) \vee \boldsymbol{F}(\boldsymbol{F}A \wedge B))$,

T7: $(\boldsymbol{P}A \wedge \boldsymbol{P}B) \Rightarrow (\boldsymbol{P}(A \wedge \boldsymbol{P}B) \vee \boldsymbol{P}(A \wedge B) \vee \boldsymbol{P}(\boldsymbol{P}A \wedge B))$,

T8: $\boldsymbol{G}A \Rightarrow \boldsymbol{F}A$, T9: $\boldsymbol{H}A \Rightarrow \boldsymbol{P}A$, T10: $\boldsymbol{F}A \Rightarrow \boldsymbol{F}(\boldsymbol{F}A)$,

T11: $(A \wedge \boldsymbol{H}A) \Rightarrow \boldsymbol{F}(\boldsymbol{H}A)$, T12: $(A \wedge \boldsymbol{G}A) \Rightarrow \boldsymbol{P}(\boldsymbol{G}A)$,

IQ1: $\forall x(A \Rightarrow B) \Rightarrow (\forall xA \Rightarrow \forall xB)$, IQ2: $(\forall xA \wedge \forall xB) \Rightarrow \forall x(A \wedge B)$,

IQ3: $\forall xA \Rightarrow A[t/x]$ (if x may appear free in A and t is free for x in A, i.e., free variables of t do not occur bound in A),

IQ4: $\forall x(A \Rightarrow B) \Rightarrow (A \Rightarrow \forall xB)$ (if x does not occur free in A),

IQ5: $\forall x_1 \ldots \forall x_n(((A \Rightarrow A) \Rightarrow B) \Rightarrow B)$ ($n \geq 0$),

RCC1: $\forall r_1(\boldsymbol{C}(r_1, r_1))$, RCC2: $\forall r_1\forall r_2(\boldsymbol{C}(r_1, r_2)\Rightarrow\boldsymbol{C}(r_2, r_1))$,
PRCC1: $\forall p_1\forall r_1\forall r_2((\boldsymbol{I}(p_1, r_1)\wedge\boldsymbol{DC}(r_1, r_2))\Rightarrow\neg\boldsymbol{I}(p_1, r_2))$,
PRCC2: $\forall p_1\forall r_1\forall r_2((\boldsymbol{I}(p_1, r_1)\wedge\boldsymbol{Pa}(r_1, r_2))\Rightarrow\boldsymbol{I}(p_1, r_2))$,
PRCC3: $\forall r_1\forall r_2(\boldsymbol{O}(r_1, r_2)\Rightarrow\exists p_1(\boldsymbol{I}(p_1, r_1)\wedge\boldsymbol{I}(p_1, r_2)))$,
PRCC4: $\forall r_1\forall r_2(\boldsymbol{PaO}(r_1, r_2)\Rightarrow(\exists p_1(\boldsymbol{I}(p_1, r_1)\wedge\boldsymbol{I}(p_1, r_2))\wedge$
$\exists p_2(\boldsymbol{I}(p_2, r_1)\wedge\neg\boldsymbol{I}(p_2, r_2))\wedge\exists p_3(\neg\boldsymbol{I}(p_3, r_1)\wedge\boldsymbol{I}(p_3, r_2))))$,
PRCC5: $\forall r_1\forall r_2(\boldsymbol{EC}(r_1, r_2)\Rightarrow\exists p_1(\boldsymbol{I}(p_1, r_1)\wedge\boldsymbol{I}(p_1, r_2)\wedge$
$\forall p_2(\neg\boldsymbol{Id}(p_2, p_1)\Rightarrow(\neg\boldsymbol{I}(p_2, r_1)\wedge\neg\boldsymbol{I}(p_2, r_2)))))$,
PRCC6: $\forall p_1\forall r_1\forall r_2((\boldsymbol{I}(p_1, r_1)\wedge\boldsymbol{TPrPa}(r_1, r_2))\Rightarrow\boldsymbol{I}(p_1, r_2))$,
PRCC7: $\forall p_1\forall r_1\forall r_2((\boldsymbol{I}(p_1, r_1)\wedge\boldsymbol{NTPrPa}(r_1, r_2))\Rightarrow\boldsymbol{I}(p_1, r_2))$,
RC1: $\forall p_1\forall p_2(\boldsymbol{Arc}(p_1, p_2)\Rightarrow\boldsymbol{Reachable}(p_1, p_2))$,
RC2: $\forall p_1\forall p_2\forall p_3((\boldsymbol{Reachable}(p_1, p_2)\wedge\boldsymbol{Reachable}(p_2, p_3))\Rightarrow\boldsymbol{Reachable}(p_1, p_3))$,
HC1: $\forall p_1(\boldsymbol{NH}(p_1, p_1))$, HC2: $\forall p_1\forall p_2\forall p_3((\boldsymbol{NH}(p_1, p_2)\wedge\boldsymbol{NH}(p_2, p_3))\Rightarrow\boldsymbol{NH}(p_1, p_3))$,
MC1: $\forall o_1(\boldsymbol{NS}(p_1, p_1))$, MC2: $\forall o_1\forall o_2\forall o_3((\boldsymbol{NS}(o_1, o_2)\wedge\boldsymbol{NS}(o_2, o_3))\Rightarrow\boldsymbol{NS}(o_1, o_3))$,
DC1: $\forall p_1\forall p_3(\boldsymbol{ND}(p_1, p_1, p_3))$,
DC2: $\forall p_1\forall p_2\forall p_3\forall p_4((\boldsymbol{ND}(p_1, p_2, p_4)\wedge\boldsymbol{ND}(p_2, p_3, p_4))\Rightarrow\boldsymbol{ND}(p_1, p_3, p_4))$.

Inference rules

$\Rightarrow$E: from A and $A\Rightarrow B$ to infer B (Modus Ponens),
$\wedge$I: from A and B to infer $A\wedge B$ (Adjunction),
TG: from A to infer $\boldsymbol{G}A$ and $\boldsymbol{H}A$ (Temporal Generalization),
$\forall$I: if A is an axiom, so is $\forall xA$ (Generalization of axioms).

Various relevant logic systems are defined as follows, where we use 'X | Y' to denote any choice of one from two axiom schemata X and Y:

$\mathbf{T}_{\Rightarrow} =_{df} \{\text{E1, E2, E2}', \text{E3} \mid \text{E3}''\} + \Rightarrow\text{E}$,
$\mathbf{E}_{\Rightarrow} =_{df} \{\text{E1, E2} \mid \text{E2}', \text{E3} \mid \text{E3}', \text{E4} \mid \text{E4}'\} + \Rightarrow\text{E}$,
$\mathbf{E}_{\Rightarrow} =_{df} \{\text{E2}', \text{E3, E4}''\} + \Rightarrow\text{E}$,
$\mathbf{E}_{\Rightarrow} =_{df} \{\text{E1, E3, E4}'''\} + \Rightarrow\text{E}$,
$\mathbf{R}_{\Rightarrow} =_{df} \{\text{E1, E2} \mid \text{E2}', \text{E3} \mid \text{E3}', \text{E5} \mid \text{E5}'\} + \Rightarrow\text{E}$,
$\mathbf{T}_{\Rightarrow,\neg} =_{df} \mathbf{T}_{\Rightarrow} + \{\text{N1, N2, N3}\}$,
$\mathbf{E}_{\Rightarrow,\neg} =_{df} \mathbf{E}_{\Rightarrow} + \{\text{N1, N2, N3}\}$,
$\mathbf{R}_{\Rightarrow,\neg} =_{df} \mathbf{R}_{\Rightarrow} + \{\text{N2, N3}\}$,
$\mathbf{T} =_{df} \mathbf{T}_{\Rightarrow,\neg} + \{\text{C1~C3, D1~D3, DCD}\} + \wedge\text{I}$,
$\mathbf{E} =_{df} \mathbf{E}_{\Rightarrow,\neg} + \{\text{C1~C4, D1~D3, DCD}\} + \wedge\text{I}$,
$\mathbf{R} =_{df} \mathbf{R}_{\Rightarrow,\neg} + \{\text{C1~C3, D1~D3, DCD}\} + \wedge\text{I}$,
$\mathbf{Tc} =_{df} \mathbf{T}_{\Rightarrow,\neg} + \{\text{C3, C5~C10}\}$,
$\mathbf{Ec} =_{df} \mathbf{E}_{\Rightarrow,\neg} + \{\text{C3~C10}\}$,
$\mathbf{Rc} =_{df} \mathbf{R}_{\Rightarrow,\neg} + \{\text{C3, C5~C10}\}$,
$\mathbf{TQ} =_{df} \mathbf{T} + \{\text{IQ1~IQ5}\} + \forall\text{I}$,
$\mathbf{EQ} =_{df} \mathbf{E} + \{\text{IQ1~IQ5}\} + \forall\text{I}$,
$\mathbf{RQ} =_{df} \mathbf{R} + \{\text{IQ1~IQ5}\} + \forall\text{I}$,
$\mathbf{TcQ} =_{df} \mathbf{Tc} + \{\text{IQ1~IQ5}\} + \forall\text{I}$,
$\mathbf{EcQ} =_{df} \mathbf{Ec} + \{\text{IQ1~IQ5}\} + \forall\text{I}$,
$\mathbf{RcQ} =_{df} \mathbf{Rc} + \{\text{IQ1~IQ5}\} + \forall\text{I}$.

The minimal or weakest propositional temporal relevant logics are as:

$\mathbf{T_0Tc}$ = **Tc** + {T1~T4} + TG,

$\mathbf{T_0Ec}$ = **Ec** + {T1~T4} + TG,

$\mathbf{T_0Rc}$ = **Rc** + {T1~T4} + TG.

Note that the minimal or weakest temporal classical logic $\mathbf{K_t}$ = all axiom schemata for **CML** + →E + {T1~T4} + TG. Other characteristic axioms such as T5~T12 that correspond to various assumptions about time can be added to $\mathbf{T_0Tc}$, $\mathbf{T_0Ec}$, and $\mathbf{T_0Rc}$ respectively to obtain various propositional temporal relevant logics. Various predicate temporal relevant logics then can be obtained by adding axiom schemata IQ1~IQ5 and inference rule ∀I into the propositional temporal relevant logics. For examples, minimal or weakest predicate temporal relevant logics are as follows:

$\mathbf{T_0TcQ}$ = $\mathbf{T_0Tc}$ + {IQ1~IQ5} + ∀I,

$\mathbf{T_0EcQ}$ = $\mathbf{T_0Ec}$ + {IQ1~IQ5} + ∀I,

$\mathbf{T_0RcQ}$ = $\mathbf{T_0Rc}$ + {IQ1~IQ5} + ∀I.

Now, we can obtain various three-dimensional spatio-temporal relevant logics by adding axiom schemata about region connection, point position, and motion of mobile objects into the various predicate temporal relevant logics. For examples:

$\mathbf{ST_0TcQ}$ = $\mathbf{T_0TcQ}$ + {RCC1, RCC2, PRCC1~PRCC7, RC1, RC2, HC1, HC2, MC1, MC2, DC1, DC2},

$\mathbf{ST_0EcQ}$ = $\mathbf{T_0EcQ}$ + {RCC1, RCC2, PRCC1~PRCC7, RC1, RC2, HC1, HC2, MC1, MC2, DC1, DC2},

$\mathbf{ST_0RcQ}$ = $\mathbf{T_0RcQ}$ + {RCC1, RCC2, PRCC1~PRCC7, RC1, RC2, HC1, HC2, MC1, MC2, DC1, DC2}.

4 Reasoning about Moving Objects Based on Three-Dimensional Spatio-temporal Relevant Logic

We now show that the three-dimensional spatio-temporal relevant logics proposed in Section 3 can satisfy all the requirements present in Section 2 for the fundamental logic system to underlie representing and reasoning about moving objects in three-dimensional space.

First of all, we have the following facts about relevant logics including strong relevant logics:

The strong relevant logics provide a logical validity criterion for relevant reasoning in the sense of strong relevance, i.e. for any valid reasoning based on a strong relevant logic, its premises include no irrelevant and unnecessary conjuncts and its conclusion includes no irrelevant or unnecessary disjuncts (Note that the logical validity criterion provided by traditional relevant logics is not necessarily relevant in this sense). Therefore, in the framework of strong relevant logic, if a reasoning is valid, then the strong relevance between its premises and its conclusion can be guaranteed necessarily, i.e. the logics can certainly underlie relevant reasoning in the sense of strong relevance [5, 6, 9]. On the other hand, because the strong relevant logics are free of not

only implicational paradoxes but also conjunction-implicational and disjunction-implicational paradoxes, the logical validity criterion provided by strong relevant logics is truth-preserving in the sense of conditional. Note that the logical validity criterion provided by **CML** is truth-preserving only in the sense of material implication; it is not truth-preserving in the sense of conditional. Also note that the logical validity criterion provided by traditional relevant logics is truth-preserving only in the sense of relevant implication; it is not truth-preserving in the sense of conditional. Therefore, in the framework of strong relevant logic, if a reasoning is valid, then the truth of its conclusion in the sense of conditional can be guaranteed necessarily, i.e. the logics can certainly underlie truth-preserving in the sense of conditional [5, 6, 9].

A reasoning based on any of relevant logics including strong relevant logics is ampliative but not circular and/or tautological. This is because the notion of entailment (conditional) that plays the most intrinsic role in any reasoning is represented in relevant logics by a primitive intensional connective satisfying the Wright-Geach-Smiley criterion, i.e. to come to know the truth of an entailment without coming to know the falsehood of its antecedent or the truth of consequent [1, 2, 5, 6, 9, 15, 18].

All relevant logics including strong relevant logics reject the principle of Explosion, and therefore, they are paraconsistent but not explosive [1, 2, 5, 6, 9, 15, 18]. All relevant logics can certainly underlie paracomplete and paraconsistent reasoning.

Now, although the three-dimensional spatio-temporal relevant logics are conservative extensions of strong relevant logics, they extended temporal relevant logics and/or strong relevant logics by introducing only predicates and axiom schemata about solid-region connection, point position, and motion of mobile objects but nothing about the classical account of validity and the material implication. Therefore, any reasoning based on the three-dimensional spatio-temporal relevant logics must be relevant and truth-preserving in the sense of conditional as well as ampliative. The logics can also underlie paracomplete and paraconsistent reasoning in the way as the same as relevant logics.

On the other hand, the predicates and axiom schemata about solid-region connection and point position provide the means to represent and reason about three-dimensional geometric relationships among three-dimensional geometric regions and points. The temporal operators and related axiom schemata provide the means to represent and reason about those propositions and/or formulas whose truth-values may depend on time. The predicates and axiom schemata about motion of mobile objects provide the means to represent and reason about motion relationships among mobile objects in a three-dimensional geometric space. Therefore, the three-dimensional spatio-temporal relevant logics can also underlie three-dimensional spatial reasoning, temporal reasoning, and three-dimensional spatio-temporal reasoning. In particular, the logics can also underlie anticipatory reasoning about mobile three-dimensional geometric objects. An anticipatory reasoning is a reasoning to draw new, previously unknown and/or unrecognized conclusions about some future event or events whose occurrence and truth are uncertain at the point of time when the reasoning is being performed. By using temporal operators and predicates about motion of mobile objects, one can easily represent and reason about future situations of mobile objects moving in a three-dimensional geometric space. Consequently, the three-dimensional spatio-temporal relevant logics provide an explicit way to underlie representing and reasoning about qualitative knowledge concerning time, space, motion, speed, and distance.

5 Concluding Remarks

We have proposed the three-dimensional spatio-temporal relevant logics as the fundamental logic system to underlie representing and reasoning about moving objects in three-dimensional space. The logics are obtained by introducing predicates and axiom schemata about solid-region connection, predicates and axiom schemata about point position, and predicates and axiom schemata about motion of mobile objects into temporal relevant logics. The notion of distance are defined by predicates about point position and adjacency, predicates about movement of mobile objects, and temporal operators. The logics can underlie relevant and truth-preserving reasoning in the sense of conditional, ampliative reasoning, paracomplete and paraconsistent reasoning, three-dimensional spatial reasoning, temporal reasoning, and three-dimensional spatio-temporal reasoning. To our knowledge, no other logic proposed for spatio-temporal reasoning has these advantages.

The three-dimensional spatio-temporal relevant logics also provide a foundation for constructing more powerful logic systems to underlie other reasoning issues in various intelligent systems dealing with mobile three-dimensional geometric objects. For examples, we can add epistemic operators and related axiom schemata into the logics in order to represent and reason about cognitive and/or epistemic processes of human being about mobile three-dimensional geometric objects and their motions.

We are working on applying the three-dimensional spatio-temporal relevant logics to developing anticipatory reasoning-reacting systems [7] dealing with mobile objects moving in three-dimensional space, e.g., anticipatory reasoning-reacting systems for air traffic control, train control, expressway control, and so on [10].

A future work that is important in theory is to establish a semantic (model) theory of the three-dimensional spatio-temporal relevant logics.

References

1. Anderson, A.R., Belnap Jr., N.D.: Entailment: The Logic of Relevance and Necessity, vol. I. Princeton University Press, Princeton (1975)
2. Anderson, A.R., Belnap Jr., N.D., Dunn, J.M.: Entailment: The Logic of Relevance and Necessity, vol. II. Princeton University Press, Princeton (1992)
3. Bennett, B., Cohn, A.G., Wolter, F., Zakharyaschev, M.: Multi-Dimensional Modal Logic as a Framework for Spatio-Temporal Reasoning. Applied Intelligence 17(3), 239–251 (2002)
4. Burgess, J.P.: Basic Tense Logic. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, 2nd edn., vol. 7, pp. 1–42. Kluwer Academic, Dordrecht (2002)
5. Cheng, J.: A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery. In: Kawaguchi, E., Kangassalo, H., Jaakkola, H., Hamid, I.A. (eds.) Information Modelling and Knowledge Bases XI. Frontiers in Artificial Intelligence and Applications, vol. 61, pp. 136–159. IOS Press, Amsterdam (2000)
6. Cheng, J.: Automated Knowledge Acquisition by Relevant Reasoning Based on Strong Relevant Logic. In: Palade, V., Howlett, R.J., Jain, L.C. (eds.) KES 2003. LNCS, vol. 2773, pp. 68–80. Springer, Heidelberg (2003)

7. Cheng, J.: Temporal Relevant Logic as the Logical Basis of Anticipatory Reasoning-Reacting Systems. In: Dubois, D.M. (ed.) Computing Anticipatory Systems: CASYS 2003. AIP Conference Proceedings, vol. 718, pp. 362–375. American Institute of Physics (2004)
8. Cheng, J.: Spatio-temporal Relevant Logic as the Logical Basis for Spatio-temporal Information Systems. In: Proc. 17th Annual International Conference on Tools with Artificial Intelligence, pp. 270–274. IEEE Computer Society Press, Los Alamitos (2005)
9. Cheng, J.: Strong Relevant Logic as the Universal Basis of Various Applied Logics for Knowledge Representation and Reasoning. In: Kiyoki, Y., Henno, J., Jaakkola, H., Kangassalo, H. (eds.) Information Modelling and Knowledge Bases XVII. Frontiers in Artificial Intelligence and Applications, vol. 136, pp. 310–320. IOS Press, Amsterdam (2006)
10. Cheng, J., Goto, Y., Kitajima, N.: Anticipatory Reasoning about Mobile Objects in Anticipatory Reasoning-Reacting Systems. In: Dubois, D.M. (ed.) Computing Anticipatory Systems: CASYS 2007. AIP Conference Proceedings, American Institute of Physics (to appear, 2008)
11. Cohn, A.G., Bennett, B., Gooday, J., Gotts, N.M.: Representing and Reasoning with Qualitative Spatial Relations About Regions. In: Stock, O. (ed.) Spatial and Temporal Reasoning, pp. 97–134. Kluwer Academic, Dordrecht (1997)
12. Cohn, A.G., Bennett, B., Gooday, J., Gotts, N.M.: Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. GeoInformatica 1(3), 275–316 (1997)
13. Cohn, A.G., Gotts, N.M., Cui, Z., Randell, D.A., Bennett, B., Gooday, J.M.: Exploiting Temporal Continuity in Qualitative Spatial Calculi. In: Egenhofer, M.J., Golledge, R.G. (eds.) Spatial and Temporal Reasoning in Geographic Information Systems, pp. 5–24. Oxford University Press, Oxford (1998)
14. Cohn, A.G., Hazarika, S.M.: Qualitative Spatial Representation and Reasoning: An Overview. Fundamenta Informaticae 46(1-2), 1–29 (2001)
15. Dunn, J.M., Restall, G.: Relevance Logic. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, 2nd edn., vol. 6, pp. 1–128. Kluwer Academic, Dordrecht (2002)
16. Egenhofer, M.J., Golledge, R.G. (eds.): Spatial and Temporal Reasoning in Geographic Information Systems. Oxford University Press, Oxford (1998)
17. Erwig, M., Schneider, M.: Spatio-Temporal Predicates. IEEE Transactions on Knowledge and Data Engineering 14(4), 881–901 (2002)
18. Mares, E.D.: Relevant Logic: A Philosophical Interpretation. Cambridge University Press, Cambridge (2004)
19. Merz, S., Wirsing, M., Zappe, J.: A Spatio-Temporal Logic for the Specification and Refinement of Mobile Systems. In: Pezzé, M. (ed.) FASE 2003. LNCS, vol. 2621, pp. 87–101. Springer, Heidelberg (2003)
20. Renz, J.: Qualitative Spatial Reasoning with Topological Information. LNCS(LNAI), vol. 2293. Springer, Heidelberg (2002)
21. Stock, O. (ed.): Spatial and Temporal Reasoning. Kluwer Academic, Dordrecht (1997)
22. Venema, Y.: Temporal Logic. In: Goble, L. (ed.) The Blackwell Guide to Philosophical Logic, pp. 203–223. Blackwell, Oxford (2001)

The Design of LQR Controller Based on Independent Mode Space for Active Vibration Control

Jingjun Zhang, Lili He, Ercheng Wang, and Ruizhen Gao

Hebei University of Engineering, Handan, Hebei, 056038, China
santt88@163.com, helili0910@163.com, wang406312@163.com, ruizhenemail@163.com

Abstract. Piezoelectric materials are finding increasing applications in active vibration control of flexible structures. The active vibration control for the laminated plane bonded with piezoelectric sensors and actuators is researched in this paper. A set of vibration control equations, sensing equations and actuating equations are derived through using the modal theory for piezoelectric flexible structures, then the model is converted into the state space form. A LQR controller is designed based on the independent mode space control techniques. Finally, the paper takes a piezoelectric cantilever beam as an example, and the simulation results demonstrate the controller is able to suppress efficiently the flexible structure's vibration with piezoelectric sensors and actuators.

Keywords: Flexible structures, Active vibration control, Independent mode space control, Linear Quadratic Regulator (LQR).

1 Introduction

Recently, a great progress has been achieved in the development of flexible structures associated with piezoelectric materials as sensors and actuators. A flexible structure used in vibration control can be defined as a structure or structure's component with bonded or embedded sensors and actuators as well as an associated control system, which enable the structure to respond simultaneously to external stimulus exerted on it and then suppresses undesired effects or enhance desired effects [1]. In 1985, Bailey et al. [2] performed an experimental research on active vibration control using surface-bonded piezoelectric polymer actuators on a cantilever beam, their experiment has greatly inspired the study of the related field.

With respect to the control of flexible structures, conventional control methods have worked well in the past. However, new design methods are required to obtain improved performance and robustness characteristics from the structural control system in order to satisfy future design specifications. Positive position feedback (PPF) [3, 4, 5] and velocity feedback are two proven methods of structural control that work well with piezoelectric sensors and actuators. PPF offers quick damping for a particular mode, provided that the modal characteristics are well known, and is also easy to implement. Strain Rate Feedback (SRF) control has also been used for active damping of a flexible space structure [6]. SRF has a wider active damping region and can stabilize more than one mode given a sufficient bandwidth.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 649–658, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The above methods are primarily devised for single-mode vibration suppression, and have limited effectiveness for multi-mode vibration suppression. With a multiple-input and multiple- output (MIMO) control system, linear quadratic control methods are the preferred choice and can be used effectively for multi-mode vibration suppression. Linear Quadratic Regulator (LQR) design has been applied in flexible structure control applications [7, 8, 9]. The control input of LQR is designed to optimize the weighted sum of the quadratic indices of energy (control input) and performance. By adjusting the weights, LQR design can meet a specific requirement, for example, to minimize deflection of a flexible structure. The LQR control approach is well suited for the requirements of damping out the effect of disturbances as quickly as possible and maintaining stability robustness. For this reason, LQR control is utilized in the development of a flexible structure controller in this paper.

The paper designs a LQR controller based on the independent mode space control techniques to vibration suppression of flexible structures through the use of bonded piezoelectric sensors and actuators. A set of vibration control equations, sensing equations and actuating equations are derived through using the modal theory for the piezoelectric flexible structures, then the model is converted into the state space form. With a control objective of minimizing displacement, independent mode space control techniques is implemented on the flexible appendage in a cantilever beam configuration utilizing piezoelectric sensor output representing structural displacement. Induced vibration is subsequently controlled through the application of control signals to the bonded piezoelectric actuators. This paper presents the modeling and simulation results for this LQR controller which based on the independent mode space control techniques study, and the simulation results demonstrate the effectiveness of the method in this paper.

2 System Modeling

2.1 Piezoelectric Constitutive Equations

For active vibration control of flexible structures, piezoelectric materials are often used as sensors and actuators [10, 11, 12]. The physical equations of the piezoelectric materials can be expressed as follows:

$$\begin{aligned} \{\sigma\} &= [c]\{\varepsilon\} - [e]^T \{E\} \\ \{D\} &= [e]\{\varepsilon\} - [\zeta]\{E\} \end{aligned} \tag{1}$$

Where $\{\sigma\}, \{D\}, \{\varepsilon\}, \{E\}$ represent the stress tensor matrix, the electric displacement density vector, the strain tensor matrix and applied electrical field vector respectively. $[c], [e], [\zeta]$ are the coefficient matrix of the piezoelectric materials. Because the terminal conditions are ε and E, the piezoelectric constitutive equations for sensors and actuators applied into active vibration control of flexible structures are simplified as follows:

$$\begin{aligned} \sigma_x &= c_{11}\varepsilon_x - e_{31}{}^T E_z \\ D_x &= e_{31}\varepsilon_x + \zeta_{33} E_z \end{aligned} \tag{2}$$

2.2 Sensing Equations of the Piezoelectric Sensors

The external electric field will be as zero if there is not any vibration, but the piezoelectric sensors will produce electric charge while the beam has some external stimulus. From the equations (1) and (2), the output voltage of the ith piezoelectric sensor can be gained as follows:

$$V_i = k e_{31} b_s r_s \int_{x_{1i}}^{x_{2i}} \frac{\partial^2 w(x,t)}{\partial x^2} dx \tag{3}$$

The voltage is an amplified one by the electric charge amplifier. Where $i = 1, \cdots, r$; r and b_s represent the number and the width of the sensors respectively. r_s is the distance from the middle axis of the sensors to the middle axis of the smart beam; k is the amplified multiple of the electric charge amplifier; x_{1i} and x_{2i} is the coordinate value of the ith sensor. Then, the vibration of the smart beam can be observed effectively based on the equation.

2.3 Actuating Equations of the Piezoelectric Actuators

To find a dynamic model of the system, assumes that a voltage v_a is applied to the piezoelectric layer and we wish to find the transfer function from v_a to the elastic deflection of the beam at any given point along the beam. The differential equation describing the elastic deflection of the composite beam piezoelectric system is a Bernoulli-Euler beam equation with an additional term due to actuating layer. This equation is given by [13]:

$$\begin{aligned} \rho A(x) \frac{\partial^2 w(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left[EI(x) \frac{\partial^2 w(x,t)}{\partial x^2} \right] \\ = \frac{\partial^2}{\partial x^2}\left[\sum_{i=1}^{n} u V_{ai}(t)(H(x - x_{1i}) - (H(x - x_{2i})) \right] \end{aligned} \tag{4}$$

Where E, I, A and ρ represent respectively the Young's modulus, moment of inertia, cross-section area, and the linear mass density of the beam. Notice that if the piezoelectric patches do not cover the entire beam surface, then, both EI and ρA will be functions of x. However, since the piezoelectric layers are often thin by comparison with the base structure, we can assume that EI and ρA are uniform over the length of the beam. n is the number of the actuators; $V_{ai}(t)$ represents the voltage which is put on the ith piezoelectric actuator; $H(*)$ is the unit step function;

$u = -\frac{1}{2} b_p d_{31} E_p (t_b + t_p)$, obviously, u is a constant which depends on the geometry parameters of the composite system, where b_p, t_p and E_p represent the width, the thickness and the Young's modulus of the piezoelectric patches respectively; t_b is the thickness of the base structure. From the equation (4), we can conclude that the vibration of the base structure would be controlled by means of put some voltage on the actuators.

3 The Simplified Model

The boundary condition for the cantilever beam is:

$$w(0,t) = 0 \tag{5}$$

$$E_b I \frac{\partial w(0,t)}{\partial x} = 0 \tag{6}$$

$$E_b I \frac{\partial^2 w(l,t)}{\partial x^2} = 0 \tag{7}$$

$$E_b I \frac{\partial^3 w(l,t)}{\partial x^3} = 0 \tag{8}$$

To use the assumed modes method, the function $w(x,t)$ is expanded as an infinite series in the form [14]:

$$w(x,t) = \sum_{i=1}^{\infty} \psi_i(x) \eta_i(t) \tag{9}$$

Where $\psi_i(x)$ are the eigenfunctions (mode shapes) satisfying the ordinary differential equations, resulting from the substitution of (5) into (4)-(8). The mode shapes are assumed to be expressed as:

$$\phi_i(x) = A[\sin a_i x - \sinh a_i x + \frac{\sin a_i l + \sinh a_i l}{\cos a_i l + \cosh a_i l}(\cosh a_i x - \cos a_i x)] \tag{10}$$

Where $a_1 l = 1.875,\ a_2 l = 4.694,\ a_3 l = 7.855$, when $i = 4,5,\cdots,\ \infty$, $a_i l = \frac{2i-1}{2}\pi$.

Substitute the equation (9) into (3), the equations of the sensors can be gained as follows:

$$V_i = ke_{31}b_s r_s \sum_{i=1}^{\infty} \eta_i(t) \int_{x_{1j}}^{x_{2j}} \frac{d^2\psi_i(x)}{\partial x^2} dx \approx ke_{31}b_s r_s l_s \sum_{i=1}^{\infty} \eta_i(t) \ddot{\psi}_i(x_j) \quad (11)$$

Where l_s is the length of the piezoelectric patches, x_j is the midpoint abscissa of the jth piece; $\ddot{\psi}_i(x_j)$ is the mode strain of the ith rank when x equals to x_j. Transform the equation (11) into matrix form, the output voltage of the piezoelectric actuators can be written as:

$$V_{sr} = C_{rm}\eta_m \quad (12)$$

Where $V_{sr} = [V_{s1}, V_{s2}, \cdots, V_{sr}]^T$, is the output voltage vector; $C_{rm} = \begin{bmatrix} \ddot{\psi}_{11} & \cdots & \ddot{\psi}_{1m} \\ \vdots & \ddots & \vdots \\ \ddot{\psi}_{r1} & \cdots & \ddot{\psi}_{rm} \end{bmatrix}$ is the position matrix of the mode actuators. $\ddot{\psi}_{ij} = \ddot{\psi}_j(x_i)$ is the mode strain of the jth rank when x equals to $x_i, i = 1, \cdots, r; \quad j = 1, \cdots, m$. $\eta_m = [\eta_1, \eta_2, \cdots, \eta_m]^T$, is the mode coordinate vector. At least m observational sensors are needed if we want to observe all the m ranks of mode coordinates. Accordingly, make the number of the observational sensors r equals to m, the equation (12) can be written as:

$$V_{sm} = C_m \eta_m \quad (13)$$

Where C_m is a phalanx with m ranks.

Substitute the equation (9) into (4), the actuate equation can be gained as follows:

$$\begin{aligned} &\rho A \sum_{i=1}^{\infty} \psi_i(x)\ddot{\eta}_i(t) + EI \sum_{i=1}^{\infty} \psi_i^{(4)}(x)\eta_i(t) \\ &\qquad = \frac{\partial^2}{\partial x^2}\left[\sum_{j=1}^{n} uV_{aj}(t)(H(x - x_{1j}) - (H(x - x_{2j}))\right] \end{aligned} \quad (14)$$

Both sides of the equation (14) multiply the mode function $\psi_i(x)$, and calculate the integral along with the whole length of the beam, with the perpendicular of the mode function, the decoupled mode coordinate equation can be expressed as:

$$\ddot{\eta}_i(t) + \omega_i^2 \eta_i(t) = \sum_{j=1}^{n} l_a u V_{ai}(t) \ddot{\psi}_i(x_j) \qquad i = 1,2,\cdots,\infty \quad (15)$$

Where ω_i is the natural frequency of the ith rank, l_a is the length of the piezoelectric actuators. Put m ranks of modes into calculation, and transform the equation (15) into matrix form, this equation can be expressed as:

$$\ddot{\eta}_i(t) + \Lambda\eta(t) = B_{kn}V_{an} \tag{16}$$

Where $\eta = [\eta_1, \eta_2, \cdots, \eta_k]^T$, is the mode coordinate vector; $\Lambda = diag(\omega_1^2, \omega_2^2, \cdots, \omega_k^2)$ is a diagonal matrix; $V_{an} = [V_{a1}, V_{a2}, \cdots, V_{an}]^T$ is the input voltage vector to the piezoelectric actuators; $B_{kn} = \begin{bmatrix} \ddot{\psi}_{11} & \cdots & \ddot{\psi}_{1n} \\ \vdots & \ddots & \vdots \\ \ddot{\psi}_{k1} & \cdots & \ddot{\psi}_{kn} \end{bmatrix}$ is the position matrix of the mode actuators. Where $\ddot{\psi}_{ij} = \ddot{\psi}_i(x_j)$.

4 Active Vibration Controller Designs

The several front ranks of the modes play an important role in the vibration control of the smart beam. So, we can only put the several front ranks into control. All of the modes can be divided into two parts: the controlled modes and the rest modes, and the controlled modes equation can be written as follows:

$$\ddot{\eta}_m(t) + \Lambda_m\eta_m(t) = B_{mn}V_{an} \tag{17}$$

Where $\eta_m(t) \in R^M$ are the controlled modes; $\Lambda = diag(\omega_1^2, \omega_2^2, \cdots, \omega_k^2)$ is a diagonal matrix constituted with the front m ranks of the natural frequency's square; $B_{mn} \in R^{M\times N}$ is the mode strain matrix; $V_{an} = [V_{a1}, V_{a2}, \cdots, V_{an}]^T$ is the input voltage vector. At least m actuators are needed if we want to control all the m ranks of modes. Accordingly, make the number of the actuators equals to the ranks of the controlled modes. Then, the equation (17) can be expressed as:

$$\ddot{\eta}_m(t) + \Lambda_m\eta_m(t) = B_mV_{am} \tag{18}$$

Where B_m is a phalanx with m ranks; V_{am} is the input voltage vector. From the equations (13) and (18), a state space model of the system is found to be:

$$\begin{aligned} \dot{x}_m &= Ax_m + BV_{am} \\ y_m &= Cx_m \end{aligned} \tag{19}$$

Where $x_m = [\eta_1, \eta_2, \cdots, \eta_m; \dot{\eta}_1, \dot{\eta}_2, \cdots, \dot{\eta}_m]^T$, is the state space vector, $y_m = [V_{s1}, V_{s2}, \cdots, V_{sm}]^T$, is the output state vector; $A = \begin{bmatrix} 0 & I_m \\ -\Lambda_m & 0 \end{bmatrix}$, is the coefficient matrix, $B = [0, B_m]^T$, is the input matrix; and $C = [C_m, 0]$, is the output matrix.

When some voltage is put on the piezoelectric actuators, a control force will be produced, and it can be expressed as follows:

$$BV_{am} = F \tag{20}$$

Where $F = [F_1, F_2, \cdots, F_m]^T$ is the mode control force vector. To minimize the displacement of the flexible structures, the Linear Quadratic Regulator (LQR) based on independent mode space control techniques is designed. The control voltage for the actuators is determined by the optimal control solution of the Linear Quadratic Regulator (LQR), which is an effective and widely used linear control technique. Provided the full state vector is observable, this method can be employed to meet specific design and performance criteria. A quadratic cost function is used to minimize the performance index. The general form for the LQR is:

$$J = \frac{1}{2}\int_0^\infty [x^T Q x + F^T R F]dt \tag{21}$$

Where both Q and R is the power matrix, and Q is the state weighting matrix, R is the control weighting matrix. The solution to the LQR problem seeks a compromise between minimum energy (control input) and best performance. Form the equation (21), we can conclude that:

$$F = -Gx \tag{22}$$

$$G = R^{-1}B^T P \tag{23}$$

Where $G = \begin{bmatrix} G_1^{(1)} & \cdots & 0 & G_2^{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & G_1^{(1)} & 0 & \cdots & G_1^{(1)} \end{bmatrix}$; P is the solution to the Riccati equation:

$$-PA - A^T P + PBR^{-1}B^T P - Q = 0 \tag{24}$$

The state vector can be gained from the equation (13):

$$x = C * V_{sm} \tag{25}$$

Where $C^* = \begin{bmatrix} C_m^{-1} & 0 \\ 0 & C_m^{-1} \end{bmatrix}$; $V_{sm} = [V_{s1}, V_{s2}, \cdots, V_{sm}]^T$ is the output voltage vector of the piezoelectric sensors. From the equations (20), (22) and (25), the input voltage of the actuators can be gained:

$$V_{am} = B^{-1}GC * V_{sm} \tag{26}$$

From the equation (26), we can conclude that the controller would reserve the output voltage of the sensors and then put it into the actuators. Based on the positive and inverse piezoelectric effect of the sensors and actuators, the vibration of the flexible structures will be controlled.

5 Numerical Examples

Now, uses the method of the paper introduced, builds a model of piezoelectric cantilever beam The geometrical size of the beam $l_b \times w_b \times h_b$ is $1000 \times 29 \times 1$ (mm), the other parameters are shown in table 1 and 2.

Table 1. Character of the aluminum material

Aluminum	EX (N/m^2)	Dens(kg/m^3)	Nuxy
SOLID45	7.5842×10^{10}	2743	0.27

Table 2. Character of the piezoelectric material

PZT-5H	Dens (kg/m^3)	Relative constant		strain constant (c/m^2)			Flexible coefficient($10^{10} N/m^2$)					
SOLID5	7700	$\varepsilon_{11}^S / \varepsilon_0$	$\varepsilon_{33}^S / \varepsilon_0$	e33	e31	e15	c11	c12	c13	c33	c44	c66
		1697.53	1468.26	23.3	-6.5	17	12.6	7.95	8.41	11.7	23	23.3

The system of the active vibration control for the piezoelectric cantilever beam is shown in fig1.

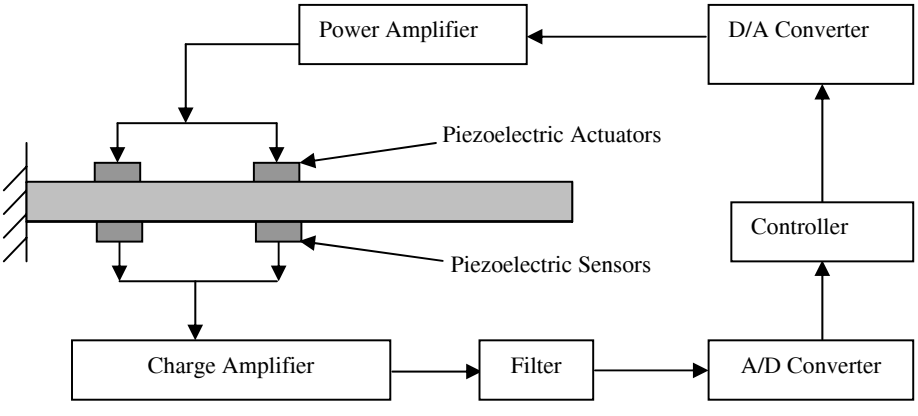


Fig. 1. Vibration control system

The piezoelectric sensors detect the displacement signals, which will be transferred to the controller though charge amplifier, Filter and A/D converter. The controller works out control signals which will be exported to the piezoelectric actuators through D/A converter and power amplifier to suppress the vibration of the cantilever beam. The first rank of the natural frequency is f1=0.81HZ, and the second rank is f2=5.1HZ. Without considering the influence of the damper in the structure system, define a primary displacement to the cantilever beam, the free vibration curves without any control are shown in fig2 and fig3.

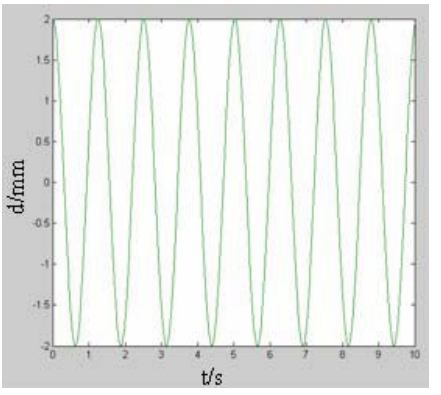


Fig. 2. Free vibration curve of the first rank mode shape without control

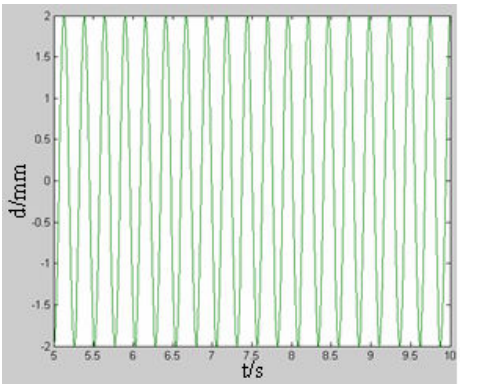


Fig. 3. Free vibration curve of the second rank mode shape without control

A LQR controller based on the independent mode space control techniques is designed in this paper, and choose the power matrixes $Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$, $R = [1]$, the first rank feedback plus matrix can be gained: $G^{(1)} = [0.0992, 1.1825]$, and the second is $G^{(2)} = [0.0049, 1.0049]$. The vibration curves under the feedback control are shown in fig4 and fig5.

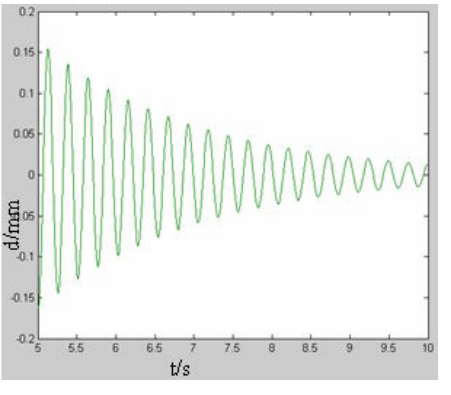


Fig. 4. Vibration curve of the first rank mode shape under control

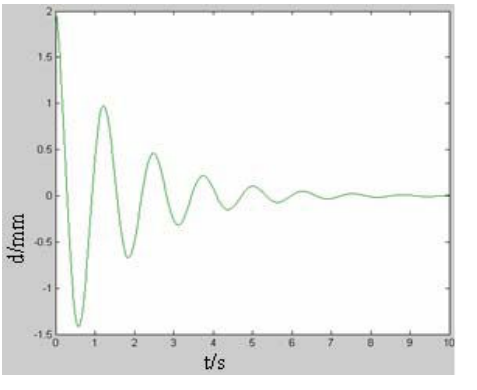


Fig. 5. Vibration curve of the second rank mode shape under control

Compare the fig2 with 4, as well as the fig3 with 5, it obvious that the application of the LQR controller which based on independent mode space control techniques suppresses the vibration of the cantilever beam effectively.

6 Conclusions

The active vibration control of a cantilever beam is studied in this paper. The paper presents an analytical investigation that focused on the development of improved vibration suppression techniques for flexible structures using piezoelectric sensors and actuators. Combining theoretical modeling with simulative developing, A set of vibration control equations, sensing equations and actuating equations are derived through using the modal theory for the piezoelectric flexible structures, then the model are converted into the state space form. A LQR controller is designed which based on the independent mode space control techniques. The simulation results demonstrate the effectiveness of the method in this paper.

Acknowledgments. The research work is supported by the Hebei natural science foundation (E2008000731) and the Hebei education department scientific research project (2006107).

References

1. Dong, X.-J., Meng, G., Peng, J.-C.: Vibration control of piezoelectric smart structures based on system identification technique, Numerical simulation and experimental study. Journal of Sound and Vibration 297, 680–693 (2006)
2. Bailey, T., Hubbard, J.E.: Distributed Piezoelectric-Polymer active vibration control of a cantilever beam. Guidant and control 8(5), 605–611 (1985)
3. Goh, C.J., Caughey, T.K.: On the Stability Problem Caused by Finite Actuator Dynamics in the Collocated Control of Large Space Structure. Int. J. Control 41(3), 787–802 (1985)
4. Fanson, J.L., Caughey, T.K.: Positive Position Feedback Control for Large Space Structure. AIAA Journal 28(4), 717–724 (1990)
5. Agrawal, B.N., Bang, H.: Adaptive Structure for Large Precision Antennas, 4.9. In: Congress of the Intemational Astronautical Federation (Jerusalem, Israel) (1994)
6. Newman, S.M.: Active Damping Control of a Flexible Space Structure Using Piezoelectric Sensors and Actuators, Master Thesis, U.S. Naval Postgraduate School (1992)
7. Won, C.C., Sulla, J.L., Sparks, D.W., Belvin, W.K.: Application of Piezoelectric Devices to Vibration Suppression. Journal of Guidance, Control, and Dynamics 17(6)
8. Agrawal, B.N.: Spacecraft Vibration Suppression Using Smart Structures. In: Fourth Intemational Congress on Sound and Vibration, St. Petersburg, Russia, June 1996, pp. 563–570 (1996)
9. Harrington, W.B.: Optimal Linear Quadratic Gaussian Controller Design for a Flexible Spacecraft Simulator, M.S. Thesis, Naval Postgraduate School (December 1995)
10. Halim, Dunant.: Vibration analysis and control of smart Structures. Australian, University of Newcastle (2003)
11. Huang, S.l., Tao, B.Q., Sheng, Y.P.: Intelligent structure systems: dream, reality and future. China mechanical engineering (2000)
12. Sung, Y.-G.: Modelling and control with piezoactuators for a simply supported beam under a moving mass. Journal of sound and vibration (2002)
13. Aldrihem, O.J.: Optimal size and location of piezoelectric actuator/sensors practical consideration. Journal of Guidance Control and Dynamics 23(3), 509–515 (2000)
14. Fraser, A.R., Daniel, R.W.: Perturbation Techniques for flexible Manipulators. Kluwer Academic Publishers, Massachusetts (1991)

A Multiagent Genetic Particle Swarm Optimization

Lianguo Wang[1,2], Yi Hong[1], Fuqing Zhao[1], and Dongmei Yu[1]

[1] Lanzhou University of Technology, Gansu 730030, China
[2] Gansu Agricultural University, Gansu 730070, China

Abstract. The efforts of this paper are proposing a multi-agent genetic particle swarm optimization algorithm (MAGPSO) by introducing the multi-agent system to the particle swarm optimization(PSO) algorithm. Through the competition and cooperation operation with its neighbors, the neighborhood random crossing operation within its neighboring area, the mutation operation, and combining the evolutionary mechanism of the PSO algorithm, every individual senses local environment unceasingly, and affects the entire agent grid gradually, so that it enhances its fitness to the environment. This algorithm can maintain the diversity of the swarm effectively, and improve the precision of optimization, and simultaneously, restrain the prematurity phenomenon efficiently. The results of testing three high dimension benchmark function and comparing with some optimization results of other methods illustrate this algorithm has higher optimization performance in the field of high dimension functions optimization.

Keywords: Multi-agent, Genetic, Particle swarm optimization, Competition, Cooperation, Function optimization.

1 Introduction

Particle Swarm Optimization Algorithm(PSO) is an evolutionary computation technology[1,2] which based on swarm intelligence method and was proposed by Kennedy and Eberhart in 1995. Many researchers improved PSO algorithm from different aspects (parameter setting, particle diversity, population structure, and algorithm fusion), and brought forward large numbers of PSO algorithms[3~10].

Reference [11] proposed multi-agent genetic algorithm, which introducing the agent systems to the genetic algorithm. Reference [12] proposed a multi-agent particle swarm optimization algorithm for reactive power optimization.

The paper proposes a multi-agent genetic PSO algorithm which puts the multi-agent system to the basic PSO and composes neighborhood random cross-operator and the mutation operator of genetic algorithm.Through continuously apperceiving local environment of every agent, combining neighborhood random cross-operator, mutation operator and PSO algorithm's evolutionary mechanism and gradually affecting the whole agent grid, every individual enhances its adaptation to the environment. By the testing of three high dimension benchmark functions and comparing with some optimization results of other methods, the results show that this algorithm has higher optimization performance in the field of high dimension functions optimization.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 659–668, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Basic Particle Swarm Optimization Algorithm

We suppose a particle swarm containing m particles flying (namely search) in the D-dimensional space, we can denote particle swarm with parameters as follows: $x_i=(x_{i,1},x_{i,2},\ldots,x_{i,D})$is the current position of particle i in D-dimensional space, $v_i=(v_{i,1},v_{i,2},\ldots,v_{i,D})$is the velocity for particle i in D-dimensional space, $P_i=(P_{i,1},P_{i,2},\ldots,P_{i,D})$is the optimized position searching so far of particle i, $P_g=(P_{g,1},P_{g,2},\ldots,P_{g,D})$is the optimized position searching so far of whole particle swarm. The particle updates its velocity and position in the D-dimensional space according to the following formulas[10]:

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \tag{1}$$

$$x_{id} = x_{id} + v_{id} \tag{2}$$

$$\begin{cases} v_{id} = v_{\max} & \text{if } v_{id} > v_{\max} \\ v_{id} = v_{\min} & \text{if } v_{id} < v_{\min} \end{cases} \tag{3}$$

where i=1,2…, M; d=1,2,…, D ; w is inertia weight, c_1 and c_2 are acceleration constants; r_1 and r_2 are the random numbers in range [0, 1]; [$v_{\min}$,$v_{\max}$] is the restricted area of the velocity for particle. Normally, w in the update speed formula reduces linearly from the largest weighting factor $w_{\max}$ to the minimum weighting factor $w_{\min}$:

$$w_t = w_{\max} - t\frac{w_{\max} - w_{\min}}{T} \tag{4}$$

Which, t is the current iteration number, T is the total number of iterative, $w_{\max}$=0.9, $w_{\min}$=0.4.

3 Multi-agent Genetic Particle Swarm Optimization Algorithm

3.1 Agent Grid

Multi-agent Genetic Particle Swarm Algorithm's population is a grid composed of the agents of N row and M range, there are $N\times M$ agents aggregately in the population, every individual is a particle. All of the agents exist in this grid which is called the agent grid, recorded as L. Every individual fixes up in one grid point, the agent in the row i and arrange j is represented as $L_{i,j,}$ i=1,2,…,N; j=1,2, …,M; and the agent $L_{i,j}$'s neighborhood is represented as $Lneighbors_{\,i,j} = \{L_{i1,j}, L_{i,j1}, L_{i2,j}, L_{i,j2}\}$,(i=1,2, …, N; j=1,2, …,M) ;They are the top, left, underside and right agents connect with this agent. Thereinto:

$$i1 = \begin{cases} i-1, & i \neq 1 \\ N, & i = 1 \end{cases} \quad j1 = \begin{cases} j-1, & j \neq 1 \\ j = M, & j = 1 \end{cases}$$
$$i2 = \begin{cases} i+1, & i \neq N \\ 1, & i = N \end{cases} \quad j2 = \begin{cases} j+1, & j \neq M \\ j = 1, & j = M \end{cases} \tag{5}$$

Every individual can only interact mutually with its neighborhood. Agent grid can be denoted as Fig 1. Every circle denotes a agent, the number in the circle denotes the position of agent in the grid, and the line denotes that agents can interact with others mutually.

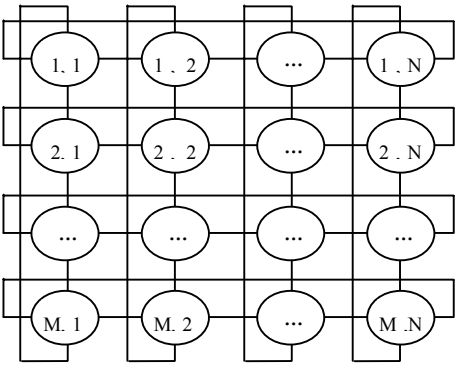


Fig. 1. Agent Grid

3.2 Agent Energy

We suppose one agent randomly is α, it is equivalent to a particle of PSO, it denotes one candidate solution of the preparative optimization function, its energy equals target function value. The aim of α is to decrease its fitness to optimum of its ability under it is satisfied with running condition's confine. For the sake of implementing its aim, the agent will make response to corresponding action according to the environment, and decrease its fitness maximally.

3.3 Multi-agent Genetic Particle Swarm Optimization Algorithm Operator Design

3.3.1 Neighborhood Competition Operator

Firstly, we find out the least fitness agent $Lmin_{i,j}$= (m_1, m_2,…, m_n) in the neighborhood of $L_{i,j}$=(l_1,l_2,…,l_n), if $L_{i,j}$'s fitness is less than $Lmin_{i,j}$'s fitness, we can keep on reserving it in the grid, otherwise, it will be replaced by $Lmin_{i,j}$. Accordingly, it reflects the competition in agents. According to formula (6), the new agent produced by $Lmin_{i,j}$ is $Lnew_{ij}$=(e_1, e_2,…, e_n), There into, U(-1,1) is the random in (-1,1), and $(\underline{x}_k,\overline{x}_k)$ is the searching part of the normal route, $k=1,2,\ldots,n$. Then, we replace $L_{i,j}$ with $Lnew_{i,j}$.

$$e_k = \begin{cases} \underline{x}_k , & \text{if } (m_k + U(-1,1)\times(m_k - l_k)) < \underline{x}_k \\ \overline{x}_k , & \text{if } (m_k + U(-1,1)\times U(m_k - l_k)) > \overline{x}_k \\ m_k + U(-1,1)\times U(m_k - l_k)) & \text{else} \end{cases} \tag{6}$$

From formula (6), we can see this occupied strategy is a heuristic crossover, in this operator, if $L_{i,j}$ is the loser in the operator, it not only still keeps on its former useful

information, but also amply absorbs its neighbor $Lmin_{i,j}$'s useful information, so its fitness decreases more.

3.3.2 Neighborhood Random Cross-Operator

Exchange a portion of neighborhood extremum with the same amount of selected particles randomly to form new particles, these new particles will replace the former particles. The predigested Neighborhood random cross-operator[13] is defined as follows: x_i= ($x_{i1,}$ $x_{i2,}$ …, $x_{iD,}$) is the current position of particle i in D-dimension space, L_{best}=($l_{best,1}$,$l_{best,2}$,…, $l_{best,D}$) is optimal extremum in the neighborhood of particle i.Firstly, we select a few units randomly from optimal extremum L_{best}=($l_{best,1}$,$l_{best,2}$,…, $l_{best,D}$) of particle i's neighborhood to exchange with the same amount of units of particle i's position x_i= ($x_{i1,}$ $x_{i2,}$ …, $x_{iD,}$), we will get crossover particle i', then we can compute its fitness. With repeating this process 4 to 5 times, then we can select optimal fitness particle i' to replace the particle i, and the speed of particle i keeps fixedness. The current position x_i'= (x_{i1}', x_{i2}', …, x_{iD}') of particle i' in the D dimension is produced by formula(7).

$$x_{ik}' = \begin{cases} x_{ih}, & U(0,1) \le pc \\ l_{best,h}, & U(0,1) > pc \end{cases} \quad k = (1,2,\cdots,D), h = [U(0,1) \times D + 1] \tag{7}$$

There into, pc is the crossover probability of k dimension unit, $U(0,1)$ is the random number of [0, 1], [ ] denotes computation integer operator, h is the position of the neighborhood extremum L_{best}'s unit which exchanged with particle i of k dimension.

3.3.3 Mutation Operator

According to formula (8), we can produce a new agent $Lnew_{i,j}$=(e_1,e_2,…,e_n) :

$$e_k = \begin{cases} l_k, & \text{if } U(0,1) < \dfrac{1}{D} \\ l_k + G(0,1/t), & \text{else} \end{cases} \tag{8}$$

There into, k is the dimension (k=1, 2,… , D), t is the current evolutionary algebra, $U(0,1)$ is equably distributing random numbers, $G(0,1/t)$ is gauss distributing random numbers. We replace $L_{i,j}$ with agent $Lnew_{i,j}$.

3.3.4 PSO Evolutionary Operator

We adopt local PSO model, and we replace the $P_{g,d}$ in the basic particle swarm algorithm formula (1) with the local best extremum unit $l_{best,d}$. Each individual in the agent grid uses their extremum and neighborhood extremum to evolve.

3.4 Multi-agent Genetic Particle Swarm Optimization Algorithm Theory

In the multi-agent genetic particle swarm algorithm, every individual agent has definite local character, every individual agent fixes up into the agent grid, and only interacts and exchanges information with its neighborhood agent. Even if one certain individual has higher fitness in the population, its effect only influences its neighborhood agent stage by stage. The algorithm implements each individual's information sufficiently utilization in the population, and leads the population to evolve with multiple directions.

Accordingly, it can effectively maintain the diversity of swarm, restrain the prematurity phenomenon. At the same time, due to the neighborhood's superposition, all of the individuals need to exchange information with each other. After a few generations evolution, some eminent agent's eminent information will diffuse to the whole population stage by stage, thus the algorithm has definite global character.

On the other hand, every individual agent has fierce competition with each other, and every individual agent has certain knowledge, it can do the PSO searching through utilize its own individual extrumum and its neighborhood best extrumum, every individual agent want to acquire more resources through the competition and to be optimal individual in this neighborhood, this can enhance its fitness to environment. At the same time, each individual agent and the neighborhood extremum make a random crossover, make full use of their own partial optimal information and neighborhood extremum information, create a good agent and substitute their own, so it advance the speed of convergence of the algorithm.

So far when the global optimum value searched has not changed continuously in the T_0 generation, the algorithm produces new agent by implementing the variation operator to the agent to further increase the diversity of the swarm and improve algorithm's ability of escape the local extremum.

3.5 Multi-agent Genetic Particle Swarm Optimization Algorithm Flow

Step1: Set algorithm parameters, and construct N×M agent grid.

Step2: Initialize every individual position and speed, and compute every individual's fitness.

Step3: Solve the position of every individual neighborhood.

Step4: Update every individual particle's speed and position according to formula (1) and (2).

Step5: Compute every individual's particle fitness.

Step6: Update every individual extremum, neighborhood extremum and global extremum.

Step7: Execute cooperation and competition operator to every individual particle neighborhood.

Step8: Execute neighborhood random cross-operator according to definite probability.

Step9: Execute mutation operator according to definite conditions.

Step10: Compute every individual's particle fitness, update every individual extremum and neighborhood extremum.

Step11: Examine termination condition (usually it is the destined most evolutionary times or good enough fitness), if the condition is not satisfied, go to step 4. Otherwise, output optimal value and terminate the algorithm.

4 Simulation Experiment

The paper make a simulation experiment by solving the minimum value of three benchmark functions to evaluate the algorithm performance. The platform of the test is VC++, the machine's master frequency is P4 (1.7G).

$$f1(X)=\sum_{i=1}^{D-1}(100(x_{i+1}^2-x_i)^2+(1\text{-}x_i)^2),\quad -100\le x_i\le 100 \tag{9}$$

The formula (9) is called Rosenbrock function, non convexity, ill-conditioned unimodal function. It will arrive at global optimal point when x_i=1(i=1,2, …, D), and the optimal value is 0.

$$f2(X)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10),\quad -10\le x_i\le 10 \tag{10}$$

The formula (10) is called Rastring function, multimodal. It will arrive at global optimal point when *xi=0(i=1,2, …, D)*, and the optimal value is 0.This function has 10D local extremum.

$$f3(X)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1\qquad -600\le x_i\le 600 \tag{11}$$

The formula (11) is called Griewank function, multimodal. It is extremely difficult to find out global optimal point for this function. It will arrive at global optimal point when *xi=0(i=1,2, …, D)*, and the optimal value is 0.

In the simulation experiment, the algorithm evaluating index is: algorithm's high performance which can find out average optimum in the number of given iteration.

We use Rosenbrock function, Rastring function amd Griewank function to search optimum, *m*, the numbers of particles are 20, 40, 80 and 160. *Gmax* is seted as 1000, 1500 and 2000 generations corresponding to the dimensions 10, 20 and 30, respectively. Basic parameter intercalates c_1=2 , c_2=2; w declines linearly from 0.9 to 0.4, v_{max}=x_{max}, the random cross probability is 0.1, the probability of random unit is 0.3 in random cross operator, mutation probability is 0.02, We do 100 experiments for every function to find out the average optimal value. The average searching optimal results

Table 1. The average optimal value of Rosenbrock function

m	Dimension	Gmax	PSO	BDPSO	IPSO	MAGPSO
20	10	1000	38.8971	20.9395	10.5172	0.1353
	20	1500	86.1087	57.0700	75.7246	5.5987
	30	2000	122.6614	63.2073	99.8038	17.4258
40	10	1000	15.4562	10.2536	1.2446	0.0036
	20	1500	70.3035	47.0046	8.7328	1.7204
	30	2000	106.7354	58.1062	14.7301	9.8633
80	10	1000	9.0187	6.6342	0.1922	0.0008
	20	1500	36.8774	22.8481	1.5824	0.0053
	30	2000	47.7536	26.9506	1.5364	6.6675
160	10	1000	10.6253	2.5160	0.0598	0.0000
	20	1500	41.6543	26.6508	0.4771	0.0000
	30	2000	44.7850	23.7709	0.4491	0.2376

of Rosenbrock function, Rastring function and Griewank function in 100 experiments are separately showed in table 1, table 2 and table 3. Comparing these results with the results of reference [14] and [15], we can see the optimal average value of MAGPSO is much more better than BDPSO[18] and IPSO[19], and with high precision. This sufficiently shows that this algorithm has obvious advantage in solving high dimensional function.

Table 2. The average optimal value of Rastrigrin function

m	Dimension	Gmax	PSO	BDPSO	IPSO	MAGPSO
20	10	1000	4.2348	3.4525	3.3298	0.00000
	20	1500	21.5688	20.0743	16.4137	0.00000
	30	2000	50.2165	34.6607	35.0.69	0.00000
40	10	1000	3.4653	3.1218	2.6162	0.00000
	20	1500	16.9452	14.5899	14.8894	0.00000
	30	2000	35.7322	28.1212	27.7637	0.00000
80	10	1000	2.2568	1.9775	1.7054	0.00000
	20	1500	11.9657	8.2386	7.6689	0.00000
	30	2000	32.1358	25.0690	13.8827	0.00000
160	10	1000	1.3520	1.0025	0.8001	0.00000
	20	1500	10.1557	9.4736	4.2799	0.00000
	30	2000	26.8588	23.4992	11.9521	0.00000

Table 3. The average optimal value of Griewark function

m	Dimension	Gmax	PSO	BDPSO	IPSO	MAGPSO
20	10	1000	0.0965	0.0915	0.0784	0.00000
	20	1500	0.0353	0.0202	0.0236	0.00000
	30	2000	0.0147	0.0104	0.0165	0.00000
40	10	1000	0.0986	0.0815	0.0648	0.00000
	20	1500	0.0328	0.0193	0.0182	0.00000
	30	2000	0.0173	0.0158	0.0151	0.00000
80	10	1000	0.0853	0.0701	0.00594	0.00000
	20	1500	0.0243	0.0222	0.0091	0.00000
	30	2000	0.0151	0.0086	0.0004	0.00000
160	10	1000	0.0899	0.0815	0.0507	0.00000
	20	1500	0.0296	0.0211	0.0048	0.00000
	30	2000	0.0114	0.0078	0.0010	0.00000

In order to compare the optimization performance of MAGPSO with the basic PSO', we use Rosenbrock function, Rastring function amd Griewank function to search optimum, set *Gmax* as 1000, dimension as 30, *m* as 20, and run the two algorithms 50 times respectively. The minimum average evolutional curves of the three functions are shown in figure 2, figure 3 and figure 4. In every figure, the y-axis denote the minimum average evolution value by common logarithm, and the x-axis denote the evolution generations. For avoiding function values turning into 0, we express the values with 10^{-7} as ending value. It can be found that after 200 generations MAGPSO can converge more quickly and the precision of optimization of MAGPSO is higher.

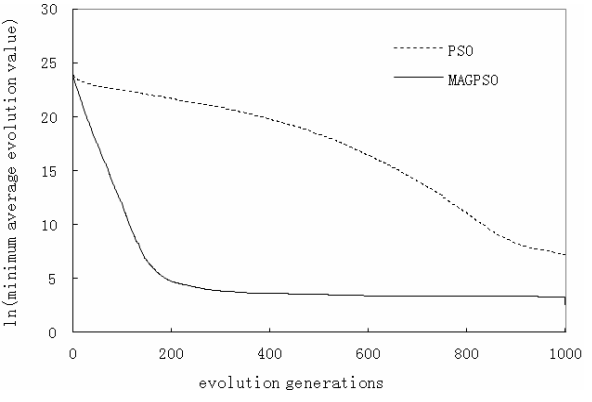


Fig. 2. The minimum average evolutional curves of Rosenbrock function

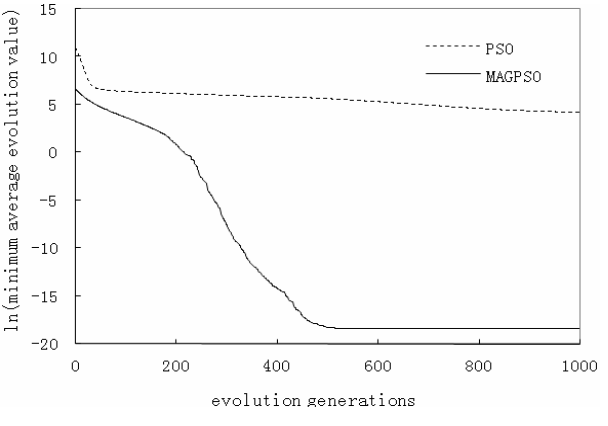


Fig. 3. The minimum average evolutional curves of Rastring function

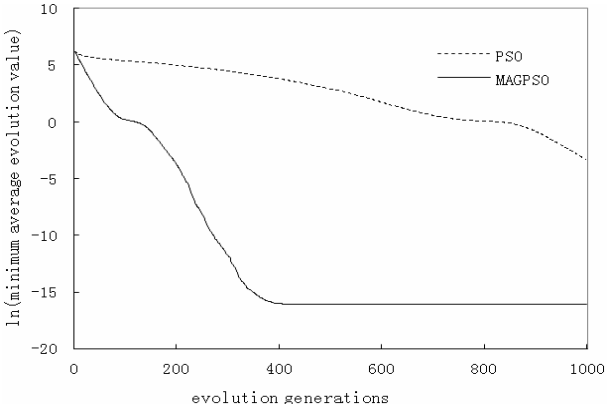


Fig. 4. The minimum average evolutional curves of Griewank function

5 Conclusion

The paper proposes a multi-agent genetic PSO algorithm(MAGPSO) by introducing the multi-agent system to the particle swarm optimization(PSO) algorithm. This algorithm can effectively maintain the diversity of the swarm, advance the speed of convergence, and increase the precision of optimization. By the testing of three high dimension benchmark functions and comparing with some optimization results of other methods, the results illustrate this algorithm has higher optimization performance in the field of high dimensional functions optimization and has a good prospect of application.

References

1. Kenned, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE Int'l. Conf. on Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
2. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proc. of the 16th International symposium on Micro Machine and Human Science, pp. 39–43. IEEE, Nagoya (1995)
3. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE World Congress on Computational Intelligence, pp. 69–73 (1998)
4. Clerc, M.: The swarm and the Queen: Towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation, pp. 1951–1957. IEEE Service Center, Piscataway (1999)
5. Angeline, P.J.: Evolutionary optimization versus particle swarm optimization: Philosophy and performance difference, C. In: Proceedings of the 7th Annual Conference on Evolutionary Programming, pp. 601–610. Springer, Gemany (1998)
6. Lovbjerg, M., Rasmussen, T.K., Krink, T.: Hybrid particle swarm optimization with breeding and subpopulations. In: Proceedings of the third Genetic and Evolutionary computation conference, vol. 1, pp. 469–476 (2001)

7. Shi, Y., Eberhart, R.: Fuzzy adaptive particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, Korea (2001)
8. Kennedy, J., Eberhart, R.: A discrete binary version of the particle swarm algorithm. In: Proceedings of the World Multiconference on Systemics,Cybernetics and Informatics 1997, pp. 4101–4109. IEEE Service Center, Piscataway (1997)
9. Higasshi, N., Hitoshi, I.: Particle swarm optimization with Gaussian mutation. In: Proceedings of the Congress on Evolutionary Computation, pp. 72–79 (2003)
10. Wu, Q., Wang, L.: Intelligent particle swarm optimization algorithm research and application, vol. 5, pp. 15–18. Jiangsu education publishing house, Nanjing (2005)
11. Jiao, L., Liu, J., Zhong, W.: Coevolutionary compution & multi-agent system, vol. 9, pp. 166–172. Science Press (2006)
12. Zhao, B., Cao, Y.: A multi-agent particle swarm optimization algorithm for reactive power optimization. J. Proceedings of the CSEE 25(5), 1–7 (2005)
13. Zhang, W., Zheng, J.: Optimization of neural network weight value by multi-agent genetic algorithms. Journal of Hubei Automotive Industries Institute 19(4), 34–36 (2005)
14. Wang, H., Qian, F.: Improved particle swarm optimizer with behavior of distance models. J. Computer Engineering and Applications 43(30), 30–32 (2007)
15. Jiang, Y., Hu, T., Huang, C., et al.: An improved particle swarm optimization Algorithm. J. Applied Mathematics and Computation 193(1), 231–239 (2007)

About the Computation Time of Adaptive Evolutionary Algorithms

Lixin Ding[1,2] and Jinghu Yu[3]

[1] School of Computer, China University of Geosciences,
Wuhan 430074, China
[2] State Key Lab of Software Engineering,
Wuhan University, Wuhan 430072, China
lxding@whu.edu.cn
[3] Department of Mathematics, School of Sciences,
Wuhan University of Technology, Wuhan 430070, China
yujh@wipm.ac.cn

Abstract. The computation time of general adaptive evolutionary algorithms based on finite search space is investigated in this paper. An adaptive evolutionary algorithm can be formalized as an inhomogeneous Markov chain. By using Markov property, some exact analytic expressions of the mean first hitting time corresponding to the adaptive evolutionary algorithm are obtained. The upper and lower bounds are also estimated by introducing drift analysis and Dynkin's formula. Furthermore, the convergence of a constructive adaptive $(1+1)-$EA is studied and its time complexity for a well-known toy problem is given.

1 Introduction

The computation time of evolutionary algorithms (EAs for brevity) for solving optimization problems is an important research topic in the foundations and theory of EAs, which reveals the number of expected generations needed to reach an optimal solution[1]. In the last over ten years, many works on this area have been made such as: Rudolph[2], Garnier et al[3,4], Droste et al[5,6]. It is quite worth mentioning He and Yao, who have done a series of works about the computation time and the time complexity for several kinds of EAs based on some different problems[7−12]. Recently, Ding and Yu[13−14] have also done some researches on this topic for general EAs based on the wider problem field by applying the theory of homogeneous Markov chain.

It is obvious that homogeneous Markov chain models have been used widely in the theoretical analysis of EAs. But in many real-world applications, when we use EAs to solve the concrete optimization problems, the genetic operators or parameter settings adopted in different step may be varied such that the associated Markov chains are not homogeneous again. Hence, it is important for us to study the complexity of computation time of EAs when their corresponding Markov chains are not homogeneous.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 669–679, 2008.
© Springer-Verlag Berlin Heidelberg 2008

In this paper we consider an inhomogeneous Markov chain associated with a general adaptive EA based on a finite search space. By introducing the definition of the first hitting time of EAs, some exact analytic expressions of the mean first hitting times of EAs are obtained. The other results in this paper concern with the upper and lower bounds of the mean first hitting times of EAs, which are acquired by applying Dynkin's Formula and some essential analytic techniques[15]. In addition, the convergence and the time complexity of a constructive adaptive $(1+1)-$EA are studied for the application examples of theoretical results of this paper.

The remaining parts of the paper are organized as follows. In section 2, we describe the formalization models of general adaptive EAs. In section 3, we obtain some exact analytic expressions of the mean first hitting times of adaptive EAs. In section 4, we give the upper and lower bounds of the mean first hitting times of adaptive EAs under some mild conditions. In section 5, we illustrate the applications of the theoretical results acquired in this paper to the especially designed adaptive $(1+1)-$EA based on the One-max problem. Finally, we conclude the paper with a short comments in section 6.

2 Description of the Model

In this paper, we consider the following optimization problem: Given an objective function with upper bound, $f : S \to R$, where S is a finite search space and R is the real line, a maximization problem is to find a $x^* \in S$ such that

$$f(x^*) = \max\{f(x) : x \in S\}. \tag{1}$$

We call x^* an **optimal solution** and write $f_{\max} = f(x^*)$ for convenience. If there are more than one optimal solution, then denote the set of all optimal solutions by S^*, and call it an **optimal solution set**.

The formalization model of an adaptive evolutionary algorithm with the population size N for solving the optimization problem (1) can be generally described as follows.

Step 1. Initialize, either randomly or heuristically, an initial population of N individuals, denoted it by $\xi_0 = (\xi_0(1), \cdots, \xi_0(N))$, where $\xi_0(i) \in S, i = 1, \cdots, N$, and let $k = 0$.

Step 2. Generate a new (intermediate) population by adopting the so-called adaptive genetic operators G_k (where G_k is varied with k), and denote it by $\xi_{k+1/2}$.

Step 3. Select and reproduce N individuals from population $\xi_{k+1/2}$ and ξ_k according to the designed survivor strategy T_k (where T_k is varied with k), and obtain the next population ξ_{k+1}, then go to step 2.

In the above algorithm, we always write $f(\xi_k) = \max\{f(\xi_k(i)) : 1 \le i \le N\}$, $\forall k = 0, 1, 2, \cdots$, if it does not bring confusion.

It is well known that $\{\xi_k; k \ge 0\}$ is a Markov chain taking values on **the state space** S^N, and if G_k and T_k do not depend on k, then $\{\xi_k; k \ge 0\}$ is called **homogeneous**, otherwise, if G_k and T_k depend on k, then $\{\xi_k, k \ge 0\}$ is

called **inhomogeneous**. The concept of first hitting time of a Markov chain has been widely used in various areas stretching from stochastic search problems, but in those models, almost all Markov chains are homogeneous. In this paper, we will pay attention to adaptive EAs described as above, in which the operators T_k and G_k depend on the generation number k. Our main aim is to estimate the number of generations for the adaptive EA to find an optimal solution and gain some insights of investigating the time complexity of adaptive EAs.

Let $d'(\cdot) : S \to [0, \infty]$ be a extended function, **it is called the test function defined on** S. For a population $\xi = (\xi(1), \cdots, \xi(N)) \in S^N$, define

$$d(\xi) = \min\{d'(\xi(i)) : i = 1, \cdots, N\}. \tag{2}$$

Usually, d' is chosen as the **distance** between the individual and the optimal solution(or optimal solution set). For example, we can define it by $f_{max} - f(\cdot)$ in the problem (1). Then d is also a non-negative **test function defined on** S^N and it is used to measure the distance between the population and the **optimal population (or optimal population set)**. In this paper, the optimal populations refer to those which include at least an optimal solution, while the optimal population set consists of all optimal populations. If d' is given by $f_{max} - f(\cdot)$ then the optimal population set with respect to d is defined by

$$(S_d^N)^* = \{\xi \in S^N : d(\xi) = 0\}. \tag{3}$$

For convenience, we write $C^* = (S_d^N)^*$.

Similar to common literatures, **the one-step drift** of stochastic sequence $\{\xi_k; k \geq 0\}$ at time k can be defined by

$$\triangle\left(d(\xi_k)\right) = d(\xi_{k+1}) - d(\xi_k). \tag{4}$$

The one-step drift describes the local performance of EAs. By using some techniques of the stochastic process, we can analyze the mean first hitting time on C^* under some mild conditions imposed on the one-step drift.

Let $N \geq 1$ be a fixed integer which represents the population size of EAs, $\mathbf{E}$ denote the expectation operator and $I_A(\cdot)$ be an index function on the set A, respectively. Write $Z^+ = \{1, 2, 3 \cdots\}$. Throughout this paper, we always assume that the stochastic process, $\{\xi_k; k \geq 0\}$, associated with the above adaptive EA, is a **finite inhomogeneous Markov chain**.

In the following section, we will give some exact analytic expressions of the mean first hitting times of EAs.

3 The Mean First Hitting Times

Let $(\Omega, \mathcal{F}, P)$ be the underlying probability space, which can support all randomization used in this paper. Assume that $\{\xi_k; k \geq 0\}$ is the corresponding inhomogeneous Markov chain associated with an adaptive EA described in section 2. Obviously its state space is S^N. Suppose that there are m(usually, $m = 2^n$,

where n is the length of individual strings) feasible solutions in search space S, thus we can sort all states in S^N by $s_1, s_2, \cdots, s_{m^N}$. For any $i, j = 1, \cdots, m^N$, let $p_{ij}^{(k)} = P\{\xi_{k+1} = j | \xi_k = i\}$, and let $P^{(k)} := (p_{ij}^{(k)})$, then $P^{(k)}$ represents the kth-step transition matrix. Write $q = (q_1, \cdots, q_{m^N})$ for **the starting distribution**, that is, $P\{\xi_0 = s_j\} = q_j, j = 1, 2, \cdots, m^N$.

In the beginning, we recall the definition of the optimal population. A population, $\xi^* = (\xi^*(1), \cdots, \xi^*(N))$, is called an optimal population in S^N, if $\xi^*(j) \in S^*$ for at least one $j (j \in \{1, \cdots, N\})$. **The first hitting time on** ξ^* can be defined by

$$\tau(\xi^*) = \min\{k \geq 0 : \xi_k = \xi^*\}. \tag{5}$$

It is obvious that, for any given optimal population ξ^*, there exist $i (i \in \{1, \cdots, m^N\})$ such that $s_i = \xi^*$. Let us write $P_{\xi^*}^{(k)}$ the matrix obtained from $P^{(k)}$ by deleting those elements of its the $i-$th column and the $i-$th row. And let $q_{\xi^*} = (q_1, \cdots, q_{i-1}, q_{i+1}, \cdots, q_{m^N})$.

Let $\mathbf{I}$ denote the $(m^N - 1) \times (m^N - 1)$ identity matrix and let $\mathbf{1} = (1, 1, \cdots, 1)'$ be the $(m^N - 1)$-dimension vector. Then we have

Theorem 1. *Let $\tau(\xi^*)$ be the number of generations for the EA to find the optimal population ξ^* for the first time. For the optimal population ξ^*,*

$$E[\tau(\xi^*)] = q_{\xi^*} \sum_{l \geq 1} \prod_{k=1}^{l-1} P_{\xi^*}^{(k)} \cdot \mathbf{1} \tag{6}$$

Proof. By Markov property of $\{\xi_k; k \geq 0\}$, for any $l \geq 1$, we have

$$\begin{aligned} P\{\tau(\xi^*) \geq l\} &= P\{\xi_0 \neq \xi^*, \xi_1 \neq \xi^*, \cdots, \xi_{l-1} \neq \xi^*\} \\ &= \sum_{y_0 \neq \xi^*, y_1 \neq \xi^*, \cdots, y_{l-1} \neq \xi^*} P\{\xi_0 = y_0, \cdots, \xi_{l-1} = y_{l-1}\} \\ &= \sum_{y_0 \neq \xi^*, y_1 \neq \xi^*, \cdots, y_{l-1} \neq \xi^*} P\{\xi_0 = y_0\} \times P\{\xi_1 = y_1 | \xi_0 = y_0\} \\ &\times P\{\xi_2 = y_2 | \xi_1 = y_1\} \times \cdots \times P\{\xi_{l-1} = y_{l-1} | \xi_{l-2} = y_{l-2}\} \\ &= q_{\xi^*} \prod_{k=1}^{l-1} P_{\xi^*}^{(k)} \mathbf{1}. \end{aligned}$$

Therefore

$$\begin{aligned} \mathbf{E}[\tau(\xi^*)] &= \sum_{k \geq 0} k \times P\{\tau(\xi^*) = k\} \\ &= \sum_{l \geq 1} P\{\tau(\xi^*) \geq l\} \\ &= q_{\xi^*} \cdot \sum_{l \geq 1} \prod_{k=1}^{l-1} P_{\xi^*}^{(k)} \cdot \mathbf{1}. \end{aligned}$$

This completes our proof. □

More generally, if $C^* = \{s_{i_1}, \cdots, s_{i_r}\}(\subset S^N)$ then we can define **the first hitting time on** C^* by

$$\tau(C^*) = \min\{k \geq 0 : \xi_k \in C^*\}. \tag{7}$$

Let $P_{C^*}^{(k)}$ be the $(m^N - r) \times (m^N - r)$ matrix obtained from $P^{(k)}$ by deleting those elements of its the i_1-th ,$\cdots$, the i_r-th columns and the i_1-th ,$\cdots$, the i_r-th rows, and $q_{C^*} = (q_1, \cdots, q_{i_1-1}, q_{i_1+1}, \cdots, q_{i_r-1}, q_{i_r+1}, \cdots, q_{m^N})$, we can get the following conclusion

Theorem 2. *Let $\tau(C^*)$ be the number of generations for the population of the EA to reach the optimal population set C^* for the first time. For the optimal population set C^*, if $\mathbf{I} - P_{C^*}$ is invertible, then*

$$\mathbf{E}[\tau(C^*)] = q_{C^*} \sum_{l \geq 1} \prod_{k=1}^{l-1} P_{C^*}^{(k)} \cdot \mathbf{1} \tag{8}$$

where $\mathbf{I}$ is a $(m^N - r) \times (m^N - r)$ identity matrix and $\mathbf{1} = (1, 1, \cdots, 1)'$ is a $(m^N - r)$-dimension vector.

The proof of Theorem 2 is similar to Theorem 1, so we omit it here.

Although the above results have great significance in theory, it is always not easy to calculate the mean and in practice, the estimation of the upper and lower bounds related to the problem size is more useful. Therefore, we will further consider the upper and lower bounds of the mean first hitting times on C^* in the following section.

4 The Upper and Lower Bounds

Note that the sequence $\{d(\xi_k) : k = 0, 1, 2, \cdots\}$ generated by the above adaptive EA is also an inhomogeneous Markov chain, where $d(\cdot)$ is defined in (2). By (7), the first hitting time on C^* with respect to the test function $d(\cdot)$ is also defined by

$$\tau(C^*) = \min\{k \geq 0 : \xi_k \in C^*\}. \tag{9}$$

We will impose some constraints on the one-step drift $\triangle(d(\xi_k))$ in order to obtain the upper and lower bounds of $\mathbf{E}[\tau(C^*)|\xi_0 = X]$. Some other marks and definitions should be stated aforehand.

Let $\{\mathcal{F}_n^{\xi}, n \geq 0\}$ be the σ-algebra given by $\xi_0, \xi_1, \cdots, \xi_n$. By Proposition 3.4.4 in [15], $\tau(C^*)$ is a **stopping time** with respect to $\sigma-$algebra sequence $\{\mathcal{F}_n^{\xi} : n \geq 0\}$.

For any $C \subset S^N$, define $\sigma_C = \min\{n \geq 1 : \xi_n \in C\}$, which is **the first return time on** C. In homogeneous case, **Dynkin's Formula**[15] was usually used to study the upper bound of the mean first return time by controlling the one-step average increment. In this paper, we will use it to estimate the upper and lower bounds of $\tau(C^*)$.

For stopping time $\tau(C^*)$ (τ for brevity in the following)defined in (9), we write

$$\tau^n = \min\{\tau, n, \inf\{k \geq 0 : d(\xi_k) \geq n\}\}, \quad \forall n \in Z^+.$$

Obviously, τ^n is also a stopping time.

Before giving our main results of this section, we first introduce some fundamental conclusions on the drift analysis of Markov chains in [15], which will be used in our proofs essentially.

Lemma 1. *(Dynkin's Formula) For any $X \in S^N$ and $n \in Z^+$,*

$$\mathbf{E}[d(\xi_{\tau^n})|\xi_0 = X] = d(X) + \mathbf{E}[\sum_{i=1}^{\tau^n}(\mathbf{E}[d(\xi_i)|\mathcal{F}_{i-1}^{\xi}] - d(\xi_{i-1}))|\xi_0 = X]. \tag{10}$$

Remark 1. If d is a test function from $S^N \to [0, \infty)$, then (10) still holds for stopping time $\tau_n = \min\{\tau, n\}$ when n is large enough. In fact, the test function $d(\cdot)$ defined in (2) is non-negative bounded when the state space S^N is finite. Otherwise, one little restriction, $\sup_{X \in S^N} d(X) < \infty$, must be imposed on it.

In the following, we give the main result in this section.

Theorem 3. *Suppose that there exist constants $0 < b \leq a$ and two positive sequences $\{a_k, k \geq 1\}$ and $\{b_k, k \geq 1\}$ with $0 < a_k \leq a$ and $b_k \geq b$ for any $k \geq 1$ such that*

$$-a_k + a_k I_{C^*}(X) \leq \mathbf{E}[d(\xi_{k+1}) - d(\xi_k)|\xi_k = X] \leq -b_k + b_k I_{C^*}(X), \quad \forall X \in S^N \tag{C}$$

If $X \in C^*$, then $\mathbf{E}[\tau|\xi_0 = X] = 0$; otherwise if $X \notin C^*$ then

$$d(X)/a \leq \mathbf{E}[\tau|\xi_0 = X] \leq d(X)/b.$$

Proof. If $X \in C^*$, then $\tau = 0$. Our assertion is true for this case.

By the hypothesis, we know $E[d(\xi_{k+1})|\mathcal{F}_k^{\xi} - d(\xi_k)] \leq -b_k$ whenever $1 \leq k \leq \tau$. Hence for all n and X we have by Dynkin's Formula

$$0 \leq \mathbf{E}[d(\xi_{\tau^n})|\xi_0 = X] \leq d(X) - \mathbf{E}[\sum_{i=1}^{\tau^n} b_i|\xi_0 = X] \leq d(X) - bE[\tau^n|\xi_0 = X]. \tag{11}$$

By the Monotone Convergence Theorem, it follows that

$$E[\tau|\xi_0 = X] \leq d(X)/b.$$

Note that if $\omega \in \{\xi_k = X\}$, then we have

$$\mathbf{E}[d(\xi_{k+1})|\mathcal{F}_k^{\xi}](\omega) = \mathbf{E}[d(\xi_{k+1})|\xi_k = X].$$

Write $Q_k = \bigcup_{X \in S^N \backslash C^*} \{\omega : \xi_k = X\}$, then we have

$$\begin{aligned}
\mathbf{E}d(\xi_{k+1}) &= \mathbf{E}[\mathbf{E}[d(\xi_{k+1})|\xi_k]] \\
&= \int_{Q_k} + \int_{\Omega \backslash Q_k} \mathbf{E}[d(\xi_{k+1})|\xi_k] dP \\
&\leq \int_{Q_k} (d(\xi_k) - b_k) dP + \int_{\Omega \backslash Q_k} d(\xi_k) dP \\
&= \mathbf{E}d(\xi_k) - b_k P(Q_k).
\end{aligned}$$

By induction on k, we have

$$0 \leq \mathbf{E}d(\xi_{k+1}) \leq \mathbf{E}d(\xi_0) - \sum_{i=0}^{k} b_k P(Q_k), \forall k \geq 1.$$

Hence, we must have

$$P(Q_k) \to 0 \qquad as \quad k \to \infty. \tag{12}$$

Since the state space is finite, (12) implies that

$$\mathbf{E}d(\xi_k) \to 0 \qquad as \quad k \to \infty.$$

So

$$\mathbf{E}[d(\xi_k)|\xi_0 = X] = \frac{\int_{\xi_0 = X} d(\xi_k) dP}{P(\xi_0 = X)} \leq \frac{\mathbf{E}d(\xi_k)}{P(\xi_0 = X)} \to 0 \qquad as \quad k \to \infty. \tag{13}$$

By the hypotheses of Theorem 3 and Dynkin's Formula, we know that if $X \in S^N \backslash C^*$, then we have

$$\begin{aligned}
a\mathbf{E}[\tau^n|\xi_0 = X] &\geq d(X) - \mathbf{E}[d(\xi_{\tau^n})|\xi_0 = X] \\
&\geq d(X) - \mathbf{E}[d(\xi_\tau)|\xi_0 = X] - \mathbf{E}[d(\xi_n)|\xi_0 = X] \\
&= d(X) - \mathbf{E}[d(\xi_n)|\xi_0 = X], \quad \forall n \in Z^+.
\end{aligned}$$

Note that $\tau^n \uparrow \tau (n \to \infty)$. By the Monotone Convergence Theorem and (13), it follows that

$$\mathbf{E}[\tau|\xi_0 = X] \geq d(X)/a, \quad X \in S^N \backslash C^*.$$

This completes our proof. □

5 An Adaptive $(1+1)-$EA

Consider the following well-known toy problem.

$$f(x) = w_0 + \sum_{i=1}^{n} w_i x_i, \tag{14}$$

where $x = (x_1, x_2, \cdots, x_n)$ is a binary string, the weights are sorted, i.e., $w_n > w_{n-1} > \cdots > w_1 > w_0 > 0$. Obviously, there is only one global optimal solution $x^* = (1, \cdots, 1)$ for the problem (14).

A traditional $(\mathbf{1+1})-\mathbf{EA}$(the corresponding Markov chain is homogeneous) has be proved to solve the problem (14) in time $\Theta(n \ln n)$[6].

Note that the population size of $(1+1)-$EAs is 1, that is to say, a population is just an individual in a $(1+1)-$EA. So we do not differentiate the original meaning of individual and population specially if without confusion. An adaptive $(1+1)-$EA (the corresponding Markov chain is inhomogeneous) to solve the problem (14) can be described as follows.

Initialization. Chose an initial individual (population) randomly from the search space S, each individual s in space S is chosen with probability $\beta(s)$,where $\beta(s) \geq 0$ and $\sum\limits_{s \in S} \beta(s) = 1$.

Mutation. At generation $k(= 0, 1, \cdots)$, for the individual(population) $\xi_k(= (x_1, \cdots, x_n))$, flip each of its bits independently with probability $p(i,k)$ for any $i = 1, \cdots, n$. The mutated individual(population) is denoted by $\xi_k^{(m)}$.

Selection. If $f(\xi_k^{(m)}) \geq f(\xi_k)$, then $\xi_{k+1} = \xi_k^{(m)}$. Otherwise, $\xi_{k+1} = \xi_k$.

Immediately, we can get the following convergence result of the adaptive $(1+1)-$EA.

Theorem 4. *Assume that there exists* $0 < p < 1$ *such that for any* $i = 1, \cdots, n$ *and* $k \geq 0$, $0 < p(i,k) \leq p$. *Then* $\lim\limits_{n \to \infty} P\{\xi_k = x^*\} = 1$.

Proof. We construct a new Markov chain as follows:

Firstly, let $f_1, \cdots, f_m$ be the m values that f will take and let $S_i = \{x \in S : f(x) = f_i\}$. Take one element s_i in each set S_i and definition a new state set $\tilde{S} = \{s_i : s_i \in S_i, i = 1, \cdots, m\}$;

Secondly, construct a Markov chain $\{\tilde{\xi_k} : k \geq 0\}$ so that it takes values in $\tilde{S}$, and its starting distribution $\tilde{\beta}$ is defined by $P\{\tilde{\xi_0} = s_i\} = \sum\limits_{s \in S_i} P\{\xi_0 = s\}$;

Thirdly, the k-th step transition matrix $\tilde{P}^k$ of $\{\tilde{\xi_k} : k \geq 0\}$ is given by

$$\tilde{P}_{ij}^k = P\{\tilde{\xi}_{k+1} = s_j | \tilde{\xi}_k = s_i\} = \frac{\sum\limits_{x \in S_i, y \in S_j} P\{\xi_{k+1} = y | \xi_k = x\} P\{\xi_k = x\}}{\sum\limits_{x \in S_i} P\{\xi_k = x\}}$$

for $s_i \in \tilde{S}, s_j \in \tilde{S}$.

Because the above adaptive $(1+1)-$EA adopts elitist strategy, so that for any k there exists permutation matrix B such that $B\tilde{P^k}B^T$ is a lower triangle matrix, that is

$$B\tilde{P^k}B^T = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \tilde{p}_{21}^k & \tilde{p}_{22}^k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{p}_{m1}^k & \cdots & \tilde{p}_{m(m-1)}^k & \tilde{p}_{mm}^k \end{pmatrix}$$

Under the assumption we can prove by induction that

$$\lim_{k\to\infty} B\tilde{P}^1\cdots\tilde{P}^k B^T = \begin{pmatrix} 1\,0\cdots 0 \\ 1\,0\cdots 0 \\ \vdots\,\vdots\,\vdots\,0 \\ 1\,0\;\vdots\;0 \end{pmatrix}.$$

Therefore, for any starting distribution $\tilde{\beta}$, we have

$$\lim_{k\to\infty}\tilde{\beta}\tilde{P}^1\cdots\tilde{P}^k = \lim_{k\to\infty}\tilde{\beta}B^T\begin{pmatrix} 1\,0\cdots 0 \\ 1\,0\cdots 0 \\ \vdots\,\vdots\,\vdots\,0 \\ 1\,0\;\vdots\;0 \end{pmatrix}B = (0,\cdots,1,0,\cdots),$$

which means that $\{\tilde{\xi_k}; k \geq 0\}$ convergence in probability Note that $P\{\tilde{\xi_k} = x^*\} = P\{\xi_k = x^*\}$ (in fact, we can prove that $P\{\tilde{\xi_k} = s_i\} = \sum\limits_{x\in S_i} P\{\xi_k = x\}$). So we get that $\{\xi_k; k \geq 0\}$ convergence in probability. □

As an application example of theoretic results of this paper, we use Theorem 3 to estimate $\mathbf{E}[\tau(x^*)]$ for $(1+1)$-EAs when $n = 2$ and $\omega_2 > \omega_1$. In this model, we assume that each bit flips independently with probability $p(i,k) = p_k, \forall i = 1,\cdots,n$ at step k and there exist r_1, r_2 such that $0 < r_2 \leq p_k \leq r_1 < 1$.

In order to obtain the upper and lower bounds of $(1+1)-$EAs, we need to verify the condition **(C)** under the following test function

$$d(x) = 4(n-1)^2\cdot\sum_{i=1}^{n}|x_i - 1| = 4\sum_{i=1}^{n}|x_i - 1|. \tag{15}$$

Note that $x^* = \{(1,1)\}$ and $d(x)(x \in S)$ only takes three different values $0, 4, 8$. We can check that

for $x = (0,0)$

$$\begin{aligned} &\mathbf{E}[d(\xi_{k+1}) - d(\xi_k)|\xi_k = (0,0)] \\ &= -4p_k(1-p_k) - 4p_k(1-p_k) - 8(1-p_k)^2 \\ &= -8(1-p_k) \end{aligned}$$

for $x = (1,0)$

$$\begin{aligned} &\mathbf{E}[d(\xi_{k+1}) - d(\xi_k)|\xi_k = (1,0)] \\ &= -4p_k(1-p_k) \end{aligned}$$

and for $x = (0,1)$

$$\begin{aligned} &\mathbf{E}[d(\xi_{k+1}) - d(\xi_k)|\xi_k = (0,1)] \\ &= -4p_k(1-p_k) \end{aligned}$$

Obviously,

$$-8(1-r_2) \le -8(1-p_k) \le -8(1-r_1)$$

and

$$-4r_1(1-r_2) \le -4p_k(1-p_k) \le -4r_2(1-r_1).$$

Hence, for any $x \neq (1,1)$

$$\begin{aligned} &\min\{-8(1-r_2), -4r_1(1-r_2)\} \\ &\le \mathbf{E}[d(\xi_{k+1}) - d(\xi_k)|\xi_k = x] \\ &\le \max\{-8(1-r_1), -4r_2(1-r_1)\}. \end{aligned}$$

Let $a = -\min\{-8(1-r_2), -4r_1(1-r_2)\}$, $b = -\max\{-8(1-r_1), -4r_2(1-r_1)\}$, then by theorem 3, we get that

$$d(x)/a \le E[\tau|\xi_0 = x] \le d(x)/b.$$

In fact, we also get the time complexity of $\Theta(n \ln n)$ for the above adaptive $(1+1)-$EA to solve (14) with the problem size n in the extended version of this paper.

6 Comments

The adaptive evolutionary algorithms are widely applied to solve the real-world optimization problems. But in theory, it is usually a rather hard task to analysis the time computation of an adaptive evolutionary algorithm because of its inhomogeneity. This paper makes a first attempt towards this aspect by introducing some new techniques of inhomogeneous Markov chain and stochastic stability theory. It provides some general results about the time complexity of adaptive EAs, which have great importance in theory. Furthermore, some analytic techniques and methods used in this paper, which may supply the researchers in the area of EA-theory the uses of references, are fundamental or even essential for investigating the time complexity problems of adaptive EAs.

It can be expected that some concrete and refined results of computation time will appear for the definite adaptive EAs especially designed for the certain optimization problem.

Acknowledgments. This work is supported in part by the Research Fund for the Doctoral Program of Higher Education (Grant no. 20070486081) and Natural Science Foundation for Distinguished Young Scholars in Hubei Province (Grant no. 2005ABB017).

References

1. Eiben, A.E., Rudolph, G.: Theory of evolutionary algorithms: A bird's eye view. Theoretical Computer Science 229(1-2), 3–9 (1999)
2. Rudolph, G.: How mutation and selection solve long path problems in polynomial expected time. Evolutionary Computation 4(2), 194–205 (1996)

3. Garnier, J., Kallel, L., Schoenauer, M.: Rigorous hitting times for binary mutations. Evolutionary Computation 7(2), 167–203 (1999)
4. Garnier, J., Kallel, L.: Statistical distribution of the convergence time of evolutionary algorithms for long path problems. IEEE Trans.on Evolutionary Computation 4(1), 16–30 (2000)
5. Droste, S., Jansen, T., Wegener, I.: A rigorous complexity analysis of the (1 + 1) evolutionary algorithm for linear functions with Boolean inputs. Evolutionary Computation 6(2), 185–196 (1998)
6. Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1 + 1) evolutionary algorithms. Theoretical Computer Science 276(1-2), 51–81 (2002)
7. He, J., Yao, X.: Drift analysis and average time complexity of evolutionary algorithms. Artificial Intelligence 127(1), 57–85 (2001)
8. He, J., Yao, X.: From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms. IEEE Trans. on Evolutionary computation 6(5), 495–511 (2002)
9. He, J., Yao, X.: Towards an analytic framework for analyzing the computation time of evolutionary algorithms. Artificial Intelligence 145(1-2), 59–97 (2003)
10. He, J., Yao, X.: A study of drift analysis for estimating computation time of evolutionary algorithms. Natural Computing 3, 21–35 (2004)
11. He, J., et al.: A note on problem difficulty measure in black-box optimization: Classification, existence and predictability. Evolutionary Computation 15(4), 435–443 (2007)
12. Oliveto, P.S., He, J., Yao, X.: Time complexity of evolutionary for combinatorial optimization: A decade of results. International J. of Automation and Computing 4(3), 281–293 (2007)
13. Ding, L.X., Yu, J.H.: Some analysis about the time complexity of evolutionary algorithms. Neural, Parallel & Scientific Computation 14(1), 51–68 (2006)
14. Ding, L.X., Yu, J.H.: Some theoretical results about the asymptotic behaviors of evolutionary algorithms. LNCS (LNAI), vol. 4456, pp. 156–166. Springer, Heidelberg (2007)
15. Meyn, S.P., Tweedie, R.L.: Markov Chains and Stochastic Stability, 3rd edn. Springer, New York (1996)

Frequency-Domain Adaptive Filtering of Batch Dealing Used for GPS Anti-jamming

Hongtao Song, Lin Zhao, Jicheng Ding, and Ming Sun

College of Automation, Harbin Engineering University
150001, Harbin, China
Mydsht@yahoo.com.cn

Abstract. By analyzing the common method of LMS (Least-mean-square) frequency-domain adaptive filtering of batch dealing used for GPS (Global Positioning System) anti-jamming, the paper points out the shortage and gives a novel algorithm, the performance of which dose not depend on the initial parameters. Both LMS algorithm and the proposed one are first introduced in detail, and then the computer simulation results for LMS and the proposed algorithm show that the latter is better.

1 Introduction

The jamming signals in the GPS environment can take on a variety of different types, from continuous waveform (CW) and frequency modulated (FM) signals to wideband (WB) noise. The GPS receiver will lose the capability for capturing while existing a single-tone jamming as in [1]. People pay more and more attention to the issue of GPS anti-jamming and many adaptive filtering theories appear nowadays, such as frequency-domain adaptive filtering in [2,3], adaptive antenna array in [4], spatial temporal adaptive processing (STAP) in [5,6]. According to the LMS frequency-domain adaptive filtering of batch dealing for GPS anti-jamming, the paper gives a novel algorithm which is better than LMS.

2 LMS Frequency-Domain Adaptive Filtering of Batch Dealing

2.1 Algorithm and Structure

The frequency-domain adaptive filtering of batch dealing used for suppressing narrowband jammers is shown by Fig.1, and the principle is given in [7] as follows.

(1) We consider $x_{Fi}(k)$ as the reference signal $d_{Fi}(k)$, and the output is given by

$$e_{Fi}(k)= x_{Fi}(k)-\omega_{Fi}(k)x_{Fi}(k) . \quad (1)$$

(2) Weight value factor betrayed is introduced with the weight value $\omega_{Fi}(k+1)$ updated by

$$\omega_{Fi}(k+1)=\alpha\omega_{Fi}(k)+2\mu_{Fi}e_{Fi}(k)x_{Fi}^{*}(k) . \quad (2)$$

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 680–686, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Combining (1) and (2) we obtain

$$\omega_{Fi}(k+1)=\alpha\omega_{Fi}(k)2\mu_{Fi}[1-\omega_{Fi}(k)]x_{Fi}(k)x_{Fi}*\ (k)\ . \tag{3}$$

From the principle above, we know that the weight increment is proportional to the power of input signal, and the weight $\omega_{Fi}\approx 1$ and the output $e_{Fi}\approx 0$, where strong interference exists. Conversely, the weight $\omega_{Fi}\approx 0$ and the output $e_{Fi}\approx x_{Fi}$.

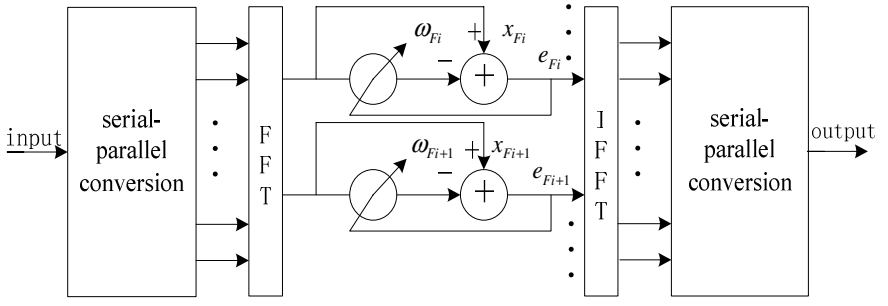


Fig. 1. Frequency-domain adaptive filtering of batch dealing

2.2 The Performance

Mean value of (3) is given by

$$E\{|\omega_{Fi}(k+1)|\}=\alpha E\{\omega_{Fi}(k)\}+2\mu_{Fi}\,E\{|x_{Fi}(k)|^2\}-E\{\omega_{Fi}(k)|x_{Fi}(k)|^2\}\ . \tag{4}$$

We suppose that $\omega_{Fi}(k)$ is independent of $x_{Fi}(k)$ and we obtain

$$E\{|\omega_{Fi}(k+1)|\}=(\alpha-2\mu_{Fi}E\{|x_{Fi}(k)|^2\})E\{\omega_{Fi}(k)\}+2\mu_{Fi}E\{|x_{Fi}(k)|^2\}\ . \tag{5}$$

The power of F_ith cell is given by

$$E\{|x_{Fi}(k)|^2\}=\sigma_{Fi}^{\ 2}\ . \tag{6}$$

We obtain

$$E\{|\omega_{Fi}(k+1)|\}=(\alpha-2\mu_{Fi}\sigma_{Fi}^{\ 2})^{k+1}E\{\omega_{Fi}(0)\}+2\mu_{Fi}\sigma_{Fi}^{\ 2}\sum_{l=0}^{k}(\alpha-2\mu_{Fi}\sigma_{Fi}^{\ 2})^{l} \tag{7}$$

In order to ensure that the limit of $E\{\omega_{Fi}(k)\}$ is constant while k tends to infinity, and $E\{\omega_{Fi}(k)\}$should be independent of $E\{\omega_{xi}(k)\}$. $\mu_{Fi}>0$ is a step size factor which satisfies $(\alpha-2\mu_{Fi}\sigma_{Fi}^{\ 2})<1$.

The convergence condition is given by

$$\mu_{Fi}\in(0,(\alpha+1)/2\sigma_{Fi}^{\ 2})\ . \tag{8}$$

$\omega_{optFi}=\sigma_{Fi}^{\ 2}/[(\alpha+1)/2\sigma_{Fi}^{\ 2}+\sigma_{Fi}^{\ 2}]$ is the optimal weight, by which we know that while the power of the spectral line $\sigma_{Fi}^{\ 2}$ is "small", the weight approach to zero, that is to say, the signal transits directly, and to the opposite, the weight tends to have the signal suppressed deeply while $\sigma_{Fi}^{\ 2}$ is "large". Here, "small" and "large" are determined by $(1-\alpha)/2\mu_{Fi}$.

2.3 Computer Simulation

The nominal electrical level of GPS signal is -160dBW, SNR is -19dB, the sampling frequency f_s = 6.138MHz , the data length for batch dealing is 1ms.

There are three single-tone jammers as $J(n)=\exp(j2\pi f_i n+\varphi)$, where $\varphi=0$. The frequencies of jammers are f_{j1}=2.0001MHz, f_{j2}=1.9801MHz, f_{j3}=1.9601MHz.

We simulate the following two scenarios with the initial parameters α=0.999, $\mu_{Fi}=(\alpha+1)/(100\sigma_{Fi}^2)$.

(1) The simulation result for the first scenario with JSR_1=40dB, JSR_2=30dB, JSR_3=20dB is given by Fig.2 to Fig.4.

(2) The simulation result for the second scenario with JSR_1=60dB, JSR_2=40dB, JSR_3=20dB is given by Fig.5 to Fig.7.

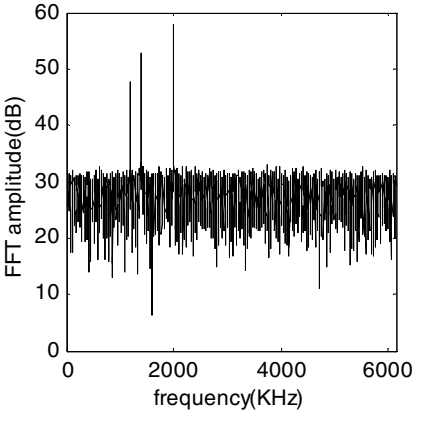


Fig. 2. Input signal spectrum for the first scenario

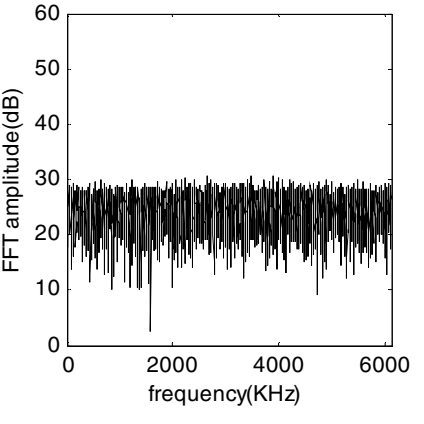


Fig. 3. Output signal spectrum based on LMS for the first scenario

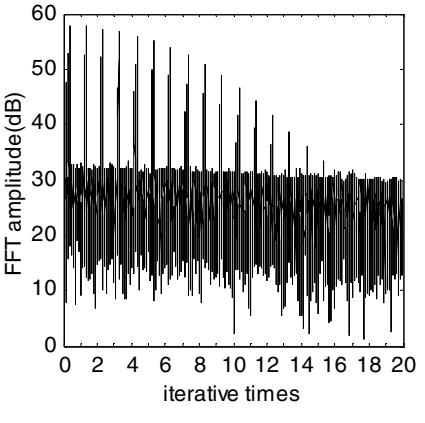


Fig. 4. Iterative figure for the first scenario

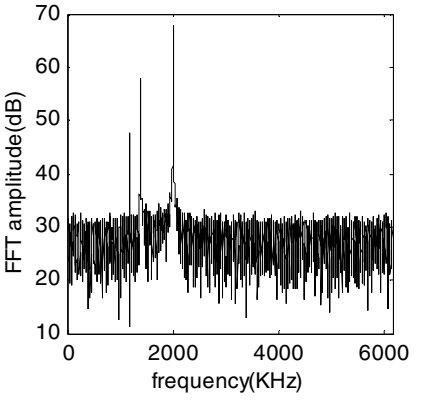


Fig. 5. Input signal spectrum for the second scenario

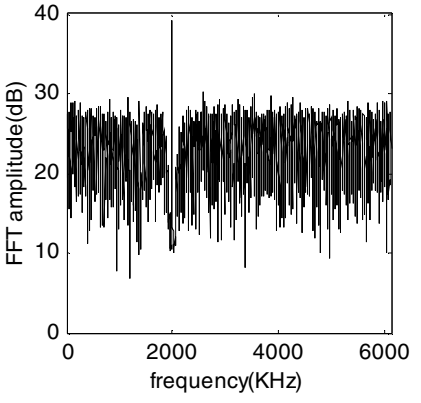


Fig. 6. Output signal spectrum based on LMS for the second scenario

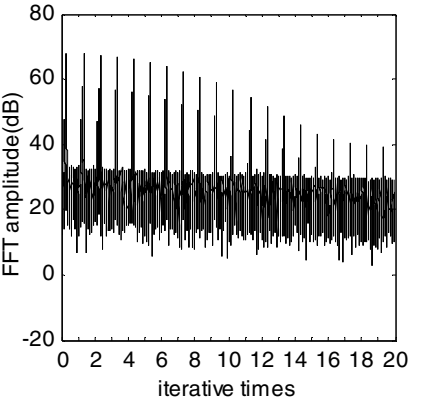


Fig. 7. Iterative figure for the second

2.4 Simulation Result

In order to simplify the simulations, we have omitted the band-pass filter which bandwidth is 2.046MHz with the center frequency 1.25MHz, and for the reason of FFT frequency resolution, we have not chosen the interference frequencies by integral multiples of 1MHz. From the anterior simulation results we can get the conclusion easily. While the initial parameters α and μ_{Fi} are chosen properly, single-tone and narrowband jammers can be well suppressed. While the power of the jammers is variational in actual application, the method can't reach a satisfied effect and we need to adjust the parameters which are important to the convergence time and it is difficult to choose in fact. In next section a new frequency-domain adaptive filtering of batch dealing is proposed, and the performance does not depend on the initial parameters.

3 The Proposed Method

3.1 Principle and Structure

From section two we know that the system performance of the LMS frequency-domain adaptive filtering of batch dealing depends on its initial parameters to a great extent, and the system will not run normally while the interference intensity change. According to this we propose a new algorithm of frequency-domain adaptive filtering of batch dealing based on the model as Fig.1, and the distinctness is that the additional weights of the latter do not depend on the initial parameters. Considering the high amplitude characteristic of FFT at the interference frequency point, we compare the FFT value of the currently frequency point with the output of the former one to obtain a currently additional weight value, the structure is given by Fig.8, and the key to the method is how to find the interference frequency.

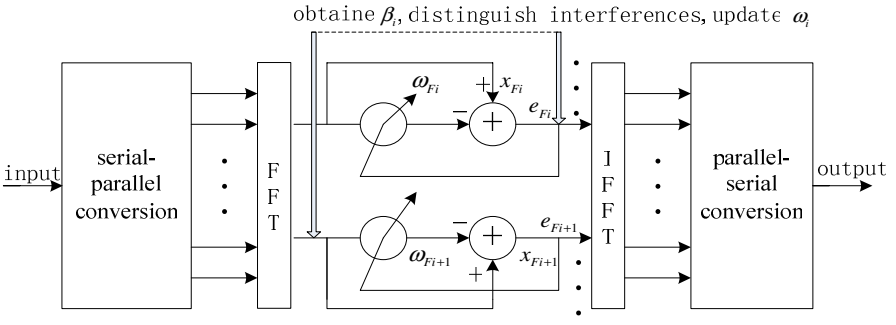


Fig. 8. Structure of the proposed method

There should be a line spectrum for a single-tone jammer, and we can consider a narrowband jammer as the summation of single-tone jammers, and it can be written as

$$J(n)=\sum_{i=-k}^{k} A_i \exp(j2\pi f_{ji} n+\varphi) \,. \tag{9}$$

We can distinguish the interference frequency component using FFT operation here, where a peak value will appear, and we also consider that the value is more than that of noise and the expected GPS signal. The novel algorithm is given as follows:

(1) Make the serial-parallel conversion and the batch processing nature of the FFT for frequency points. We suppose that the SNR of GPS signal is -19dB, and search the jamming signal frequency point by point. While the FFT value $Fx(i)$ is much larger than $e(i\text{-}1)$ which is the output of the former frequency point, we confirm that this is the jamming frequency point.

(2) For the spectral bandwidth of the GPS C/A code is 2.046 MHz, we merely search the range which is 1.023 MHz on both sides of the center frequency f_c.Out of this range we suppose $\omega(i)$ =1 however ω=0 at the first point .

(3) We define $\beta_i=Fx(i)/e(i\text{-}1)$, While $\beta_i<C$, $\omega(i)$=0,and while $\beta_i \geqq C$. $\omega(i)=(\beta_i\text{-}1)/\beta_i$ is chosen to avoid the infinite result at the same operation of the next frequency point, that is to indicate the energy at this frequency point is suppressed incompletely. The comparison value C should be chosen properly such as 100 to the knowledge of that if the FFT value at the jamming frequency point is not more than above 100 times of the FFT value of noise, the working state of the GPS receiver will not be changed, and the interference frequency points do not need to suppressed.

(4) Defining the output $e(i)=Fx(i)\text{-}\omega(i)Fx(i)$, the inverse FFT transform will be done for the subsequent processing.

3.2 Computer Simulation

The nominal electrical level of GPS signal is -160dBW, SNR is -19dB, intermediate frequency after down-converted is 1.25MHz, we suppose that there is a ideal filter before mixing, which bandwidth is 2.046MHz, the sampling frequency f_s is 6.138

MHz, the date length for batch dealing is 10ms, frequency resolution is 100Hz, a narrowband jammer exists which is written as (9) where k=3, fj0=2MHz,△f=80KHz. There are also two single-tone jammers which frequencies are 0.5MHz and1MHz, JSRs are 10dB and 5dB. The spectrums are shown by Fig.9 and Fig.10.

3.3 Simulation Result

We have omitted the band-pass filter the same as section two. According to the algorithm principle , as long as "high apices" are formed at the interference frequency points while making FFT operation, the interference frequency points can be detected accurately. In the simulation above single-tone and narrowband jammers are well suppressed with this method.

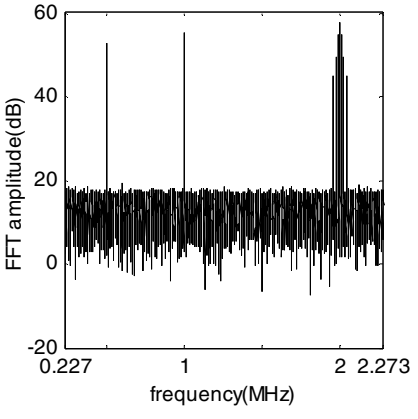


Fig. 9. Input signal spectrum based on the proposed algorithm

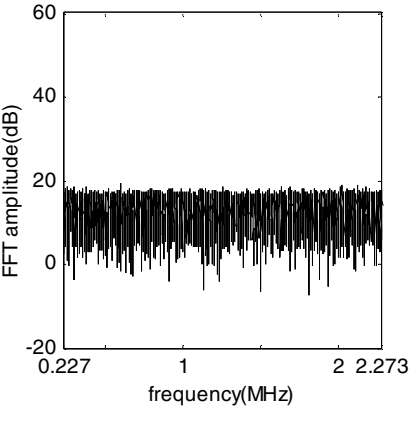


Fig. 10. Output signal spectrum based on the proposed algorithm

4 Conclusion

By analyzing the algorithm principle of the LMS frequency-domain adaptive filtering of batch dealing and the computer simulation results, we know that its performance is determined by the initial parameters and while the interference power changing we are likely to get a bad result and the system can't work normally. In this paper a simple effective algorithm which can bring us a satisfied performance is proposed, that it doesn't depend on the initial parameters in the condition with variational jammer powers interested us, and it is easily to realize on the current computer level.

References

1. Capozza, P.T., Holland, B.J., Hopkinson, T.M., Landrau, R.L.: A single-chip Narrow-band Frequency-domain Excisor for A Global Positioning System(GPS) Receiver. IEEE Journal of Solid state Circuits 35, 401–411 (2000)

2. Haykin, S.: Adaptive filter theory, 3rd edn. Prentice-Hall, Englewood Cliffs (1996)
3. Shynk, J.J.: Frequency-domain and multirate adaptive filtering. IEEE Signal Processing Magazine 9, 14–37 (1992)
4. Saulnier, G.J.: Suppression of narrowband jammers in a spread-spectrum receiver using transform-domain adaptive filtering. IEEE JSAC 10, 742–749 (1992)
5. Wen, X.-q., Han, C.-z., Han, H.: Direct Data Domain Approach to Adaptive Antennas. Acta Simulata Systematica Sinica 12, 2703–2705 (2004)
6. Di, M., Zhang, E.: Research on GPS Anti-jamming Technologies, 15–42 (2006)
7. Gong, Y.: Adaptive filter—Time-domain adaptive filter and aptitude antenna, 2nd edn. Publishing House of electronics industry, Beijing (2003)

Image Decomposition Based on Curvelet and Wave Atom

Chengwu Lu[1, 2]

[1] School of Mathematics and Statistics, Chongqing University of Arts and Sciences, Chongqing, 402160, China
[2] School of Science, Xidian University, Xi'an, 710071, China
L_chengwu@163.com

Abstract. To separate oscillating parts such as texture and noise from piecewise smooth parts, a new variational image decomposition model is presented, which well improve the novel Starck's model. The second generation curvelets and wave atoms are used to represent structure and texture respectively. The total variation semi-norm is added for restricting structure parts. The generalized homogeneous Besov norm proposed by Meyer is used to constrain noisy components. Finally, the Basis Pursuit Denoisiing algorithm is used to solve the new model. Experiments show that the approach is very robust to noise, and that can keep edges and textures stably.

1 Introduction

The task of decomposing image into simpler atoms or structure blocks is of great interest for many applications in image processing. This kind of problem has attracted considerable research attention recently. Independent Component Analysis (ICA) and sparsity methods are typically used for the separation of image mixtures with varying degrees of success. In these methods a typical assumption is made that the given image is a linear mixture of several source images of more coherent origin. For instance, a nature image is linearly mixed by three coherent components, cartoon structure image, texture image and noise image. The sparsity methods firstly choose the appropriate basis dictionary and then, to find the most sparsity representation for each component in terms of the features themselves. Mathematically, this problem is to find sparsity solutions of ill-posed equation [1].

Assume that the input image u to be processed is piecewise smooth pure cartoon structure image of size $N \times N$. We represent this image as a 1D vector of length N^2 by simple reordering. D_u denotes an over-complete basis dictionary of the space where u lies L is amount of basis in this lab. D_u is represented as an over-complete matrix M_u of size $N^2 \times L$, if each basis in the lab D_u is represented as a 1D vector of length N^2. So the sparsity representation of u in D_u can be determined by the equation, $u = M_u a_u$, where a_u denotes the unknown coefficient column vector. Due to $N^2 \ll L$ this equation has infinite solutions. Our interests focus on finding the sparsity solution of this equation. To this end, we can come down this problem to the model as follows [2].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 687–696, 2008.
© Springer-Verlag Berlin Heidelberg 2008

$$\begin{cases} a_u^{opt} = Arg \min\limits_{a_u} \|a_u\|_1 \\ u = M_u a_u \end{cases} \tag{1}$$

As the pure cartoon image, through choosing proper basis dictionary D_v (M_v is corresponding representation matrix) for the pure texture image, we can a ℓ^1 -norm model same as (1). Moreover, it is not to hard obtain the ℓ^1 -norm model for a linear mixture image "cartoon + texture".

$$\begin{cases} \left[a_u^{opt}, a_v^{opt}\right] = Arg \min\limits_{a_u, a_v} \|a_u\|_1 + \|a_v\|_1 \\ u = M_u a_u, v = M_v a_v \end{cases} \tag{2}$$

However, besides cartoon component and texture component, a practical nature image often includes the noise. We can go on above modeling so as to choose a proper basis dictionary for the noise component. Here note, all dictionaries have to choose in terms of the feature of corresponding component, but have a sparse representation for one component and non-sparse representation for other different component types.

Following the work of Starck[3], Vese, Osher and others[4-9], in this paper a improved model on Starck's model is presented, where we focus on the decomposition of a given image into a texture and natural (piecewise smooth) additive components. The improved model based on studying the recent non-adaptive sparse representation, two appropriate dictionaries be used, the second generation curvelets and the wave atoms, one for the representation of textures, and the other for the natural scene parts. In addition to, the generalized homogeneous Besov norm is introduced to constrain noise component.

2 Some Basic Theory

2.1 Second Generation Curvelet

The first generation of curvelets is based on block ridgelets [10]. Apart from the blocking effects, however, the application is limited, because the geometry of ridgelets is itself unclear as they are not true ridge functions in digital images. In this paper, we apply the second-generation discrete curvelet transform [11], [12] (DCuT), which is considerably simpler to use.

Intuitively, curvelets are waveforms that are highly anisotropic at fine scales, with effective support obeying the parabolic principle length $width \approx length^2$. Just as for wavelets, there is both a continuous and a discrete curvelet transform. A curvelet is indexed by three parameters which adopting a continuous description of the parameter space are: a scale a , $0 < a < 1$; an orientation θ , $\theta \in \left[-\pi/2, \pi/2\right)$ and a location b , $b \in \mathbb{R}^2$. Assume φ_a is some kind of directional wavelet with spatial width ~ a and spatial length,~ $\sqrt{a}$ and with minor axis pointing in the horizontal direction. At scale a , the

family of curvelets is generated by translation and rotation of a basic element φ_a, such that

$$\varphi_{a,\theta,b}(x) = \varphi_a\left(R_\theta(x-b)\right) \tag{3}$$

Here φ_a is known as continuous curvelet. Through discretizing the parameter space of φ_a, we can define the discrete curvelet as wavelet.

2.2 Wave Atom

Wave atom is some variation of wavelet packets. It may thinks as an "interpolation" between wavelet and Gabor atom [13]. Wave atom obey parabolic scaling relation, $wavelength \approx (diameter)^2$. Now we present a formal definition of 2-D wave atom. We denote wave atom by $\varphi_\mu(x)$, with subscript $\mu = (j,\mathbf{m},\mathbf{n}) = (j,m_1,m_2,n_1,n_2)$ $j,m_i,n_i \in \mathbb{Z}$ $(i=1,2)$, index a point (x_μ,ξ_μ) in phase-space, as $x_\mu = 2^{-j}\mathbf{n}$, $\xi_\mu = \pi 2^j \mathbf{m}$, $C_1 2^j \le \max_{i=1,2}|m_i| \le C_2 2^j$, where $C_1, C_2 > 0$ are two positive constants. x_μ and ξ_μ are the centers of $\varphi_\mu(x)$ in spatial and frequency domain respectively. Wave atoms then need to obey a localization condition around the phase-space point (x_μ,ξ_μ).

Definition 1. The elements of a frame of wave packets $\{\varphi_\mu\}$ are called wave atoms, if for all $M > 0$ when

$$|\varphi_\mu(x)| \le C_M \cdot 2^j \left(1+2^j|x-x_\mu|\right)^{-M} \tag{1}$$

$$|\hat{\varphi}_\mu(\xi)| \le C_M \cdot 2^j \left(1+2^j|\xi-\xi_\mu|\right)^{-M} + C_M \cdot 2^j \left(1+2^j|\xi+\xi_\mu|\right)^{-M} \tag{2}$$

Definition 1 only presents a qualitative description for wave atom with spatial-frequency location restriction. In practice, Demanet uses the strategy of frequency localization given by Villemose to construct wave atom from tensor products of adequately chosen 1D wave packets.

Denote $\mu = (j,\mathbf{m},\mathbf{n}) = (j,m_1,m_2,n_1,n_2)$, H is Hilbert transform. Firstly define an orthonormal basis $\psi_\mu^+(x_1,x_2) = \psi_{m_1}^j(x_1-2^{-j}n_1)\psi_{m_2}^j(x_2-2^{-j}n_2)$. And then a dual orthonormal basis can be defined from the Hilbert-transformed wavelet packets,
$\psi_\mu^-(x_1,x_2) = H\psi_{m_1}^j(x_1-2^{-j}n_1)H\psi_{m_2}^j(x_2-2^{-j}n_2)$. So $\{\psi_\mu\} = \{\psi_\mu^{(1)},\psi_\mu^{(2)}\}$ form

the wave atom frame in two-dimension, if denote $\psi_\mu^{(1)} = \left(\psi_\mu^+ + \psi_\mu^-\right)/2$, $\psi_\mu^{(2)} = \left(\psi_\mu^+ - \psi_\mu^-\right)/2$ and satisfy

$$\sum_\mu \left|\left\langle f, \psi_\mu^{(1)} \right\rangle\right|^2 + \sum_\mu \left|\left\langle f, \psi_\mu^{(2)} \right\rangle\right|^2 = \|f\|^2 .$$

The coefficients of two-dimension wave atom transform can be obtained as follows:

$$c_\mu = \left\langle f, \psi_\mu^{(1)} \right\rangle + \left\langle f, \psi_\mu^{(2)} \right\rangle \tag{4}$$

2.3 Depiction of Wavelet for Generalized Homogeneous Besov Space

In [1], in addition to G-space, Y. Meyer suggests using $E = \dot{B}_{\infty,\infty}^{-1}\left(\mathbb{R}^2\right)$ which is the dual space of Besov space $\dot{B}_{1,1}^1$ to model oscillatory component of image. The research shows that the generalized homogeneous Besov space is specially fit for noise which can be taken as a kind of random oscillatory component.

Assume $\psi_{j,k} = 2^j \psi\left(2^j x - k\right)$ $\left(j \in \mathbb{Z}, k \in \mathbb{Z}^2, \psi = \left\{\psi^{(1)}, \psi^{(2)}, \psi^{(3)}\right\}\right)$ be orthonormal wavelet basis in $L^2\left(\mathbb{R}^2\right)$, where $\psi^{(m)} \in C^2\ (m = 1, 2, 3)$ be tight supported. $f \in \dot{B}_{1,1}^1$ if and only if $\sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}^2} \left|\left\langle f, \psi_{j,k} \right\rangle\right| < +\infty$

Theorem 1. In above supposing, $w \in \dot{B}_{\infty,\infty}^{-1}$ if and only if the wavelet coefficients of function w belongs in Lorenz space, $\left\langle w, \psi_{j,k} \right\rangle \in \ell^\infty\left(\mathbb{Z} \times \mathbb{Z}^2\right)$.

Theorem 1 implies the equivalence wavelet depiction for Besov space $\dot{B}_{\infty,\infty}^{-1}$.

3 Image Decomposition on Generalized Homogeneous Besov Norm Restriction

Recently, Starck proposed a novel approach for image decomposition which is based on the Basis-Pursuit denoising (BPDN) algorithm, and extending results from previous work [2-3].The basic idea behind this new algorithm is to choose two appropriate dictionaries, one for the representation of textures, and the other for the natural scene parts. Both dictionaries are to be designed such that each leads to sparse representations over the images it is serving, while yielding non-sparse representations on the other content type. Starck's model is written:

$$\left\{a_u^{opt}, a_v^{opt}\right\} = Arg \min_{a_u, a_v} \|a_u\|_1 + \|a_v\|_1 + \alpha \left\|f - M_u a_u - M_v a_v\right\|_2^2 + \beta \left|M_u a_u\right|_{TV}$$

where image is decomposed into three components, structure component u whose transform coefficient denoted by a_u, texture component v whose transform coefficient denoted by a_v, and noise $f-u-v$.Now we simply remark the performance of Starck's model..

① In Starck's model, authors apply 99 curvelet basis lab to represent the construction component of image. Due to 99 curvelet is defined by overlapped local ridgelet based on multi-scale partition. This definition is very complex, but the geometric form of the orthonormal ridgelet and scale ratio relation still are not clear. In addition, 99 curvelet has redundancies with factor $16J+1$. So 99 curvelet will increase complexity of Starck's model.

② In Starck's model, texture component is modeled by DCT(Discrete Cosine Transform). However, DCT may cause blocky effect. Although one use overlapping blocks to overcome this defect in some sense, this method will produce redundancy with factor 4.

③ Noise component is not depicted by appropriate dictionary, but is obtained by fallout which is restricted by L^2 norm of remainder $f-u-v$. In terns of Meyer's oscillatory pattern theory, using L^2 norm to restrict noise is not appropriate.

Consequently, in this paper, we will modify Starck's model to overcome above mentioned limitation.

3.1 Proposing New Model

In [5], be mentioned above Starck's model is equivalently rewritten as follows:

$$\left\{u^{opt}, v^{opt}\right\} = Arg\min_{u,v} \left\|M_u^+ u\right\|_{\ell^1} + \left\|M_v^+ v\right\|_{\ell^1} + \alpha\left\|f-u-v\right\|_{L^2}^2 + \beta|u|_{TV} \quad (5)$$

where M_u^+ and M_v^+ denote generalized inverse of M_u and M_v respectively, They are over-completeness linear transform for components u and v. In practice, M_u^+ and M_v^+ are taken as curvelet transform and DCT respectively, so model (5) is rewritten further

$$\left\{u^{opt}, v^{opt}\right\} = Arg\min_{u,v} \left\|C_1 u\right\|_{\ell^1} + \left\|Dv\right\|_{\ell^1} + \alpha\left\|f-u-v\right\|_{L^2}^2 + \beta|u|_{TV} \quad (6)$$

where C_1 denotes 99 curvelet transform, D denotes DCT.

We will present modified models based on some limitations of Starck's model in two different cases. Here, we replace 99 curvelet and DCT respectively with second generation curvelet and wave atom, instead of L^2 norm, we use generalized homogenous Besov norm to restrict remainder (or noise component).

① f be noise free image, $f=u+v$, modified model,

$$\left\{u^{opt}, v^{opt}\right\} = Arg\min_{u,v} \left\|C_2 u\right\|_{\ell^1} + \left\|Av\right\|_{\ell^1} + \alpha\left\|f-u-v\right\|_{L^2}^2 + \beta|u|_{TV} \quad (7)$$

where C_2 denotes second generation curvelet transform, A wave atom. Modified model (7) has not large difference than model (6) in addition to dictionaries depicting structure and texture components. However, due to free noise, it is more appropriate to keeps L^2 norm penalty in modified model (7). The same as model (6), the new model (7) keep TV penalty term to preserve geometric structure of u.

② f be noisy image, $f = u + v + n$

Now we take $n = f - u - v$ as noise component, and apply generalized homogeneous Besov norm to restrict it. So we obtain another modified model

$$\left\{u^{opt}, v^{opt}, n^{opt}\right\} = Arg \min_{u,v} \left\|C_2 u\right\|_{\ell^1} + \left\|Av\right\|_{\ell^1} + \alpha \left\|n\right\|_E + \beta \left|u\right|_{TV} \quad (8)$$

Based on the equivalency of wavelet for E-norm, modified model (8) can be rewritten

$$\left\{u^{opt}, v^{opt}, n^{opt}\right\} = Arg \min_{u,v,n} \left\|C_2 u\right\|_{\ell^1} + \left\|Av\right\|_{\ell^1} + \alpha \left\|Wn\right\|_{\ell^1} + \beta \left|u\right|_{TV} \quad (9)$$

where W denotes wavelet transform. The third term in right hand of model (9) may be taken as using wavelet dictionary to represent noise component.

3.2 Algorithm of Modified Models

Assume C_2^{-1}, A^{-1} and W^{-1} respectively denote inverse transform of C_2, A and W, $S(\cdot)$ soft threshold function. Using the Basis-Pursuit denoising (BPDN) algorithm and equivalency between total variation and Haar wavelet [5], we can obtain the algorithm of modified model (7) and (9).

Step1: Initialize the maximal number of coefficient $L_{\max}$, the number of iterations per layer K, and threshold $s = \alpha \cdot L_{\max}$, $u = v = n = 0$;

Step 2: Perform K times iterations

Part A: Update of structure component u

① Calculate the residual: $R = f - u - v - n$

② Calculate second generation digital curvelet transform of $u + R$, $a_u = C_2(u + R)$

③ Soft threshold the coefficient a_u with the s threshold and obtain, $\hat{a}_u = S(a_u)$

④ Reconstruction u by $u = C^{-1}(\hat{a}_u)$

Part B: Update of texture component v

① Calculate the residual: $R = f - u - v - n$

② Calculate wave atom transform of $v + R$ and obtain $a_v = A(v + R)$

③ Soft threshold the coefficient a_v with s obtain, $\hat{a}_v = S(a_v)$

④ Reconstruction v by $v = A^{-1}(\hat{a}_v)$

Part C: Update of noise component n

① Calculate the residual: $R = f - u - v - n$

② Calculate (db10) wavelet transform of $n + R$ and obtain $a_n = W(n + R)$

③ Soft threshold the coefficient a_n with s and obtain, $\hat{a}_n = S(a_n)$

④ Reconstruction n by $n = W^{-1}(\hat{a}_n)$

Part D: TV consideration

① Calculate (Haar) wavelet transform of u and obtain $a_u^* = W_H(u)$

② Soft threshold the coefficient a_u^* with s and obtain, $\hat{a}_u = S(a_u^*)$

③ Reconstruction u by $u = W_H^{-1}(\hat{a}_u)$

Step 3: Update the threshold by $s = s - \alpha$;

Step 4: If $s > \alpha$, return to Step 2. Else, finish.

***Remark*:** Part C in Step 2 is optional, which does not perform when f is the noiseless image and let $n \equiv 0$.

4 Numerical Experiments

To show the validity of proposed models, we apply it and Starck's model to perform decomposition for noise free image and denoising for noisy image. In our experiments, Initialize the maximal number of coefficient $L_{\max} = 1024$, the number of iterations per layer $K = 130$, wavelet is choosed as db10 in Part C. Theoretically, both parameters α and β may be determined adaptively by certain optimal algorithm. But this procedure will need additional calculation. As our start point consists in comparing modified model with Starck's model, we will not involve algorithm to determine optimal parameter, here which are obtain experientially. In addition, to clearly observe texture component, which are obtained by $v + 150$.

(1). **Decomposition for Noise Free Image:** Using Starck's model and modified model (7) to separate noise free Barbara image into structure and texture components. Where take $\alpha = 0.4$, $\beta = 0.025$. In this case, the most remark difference of both models consist in dictionary which be used to depict texture components, where local cosine basis for Starck's model and wave atom for our modified model (7). As experimental results in Fig. 1, modified model (7) separates more texture than Starck's model, for instance, some texture information on table cloth are lost when applying Starck's model, however, model (7) can extract the most texture including on table cloth.

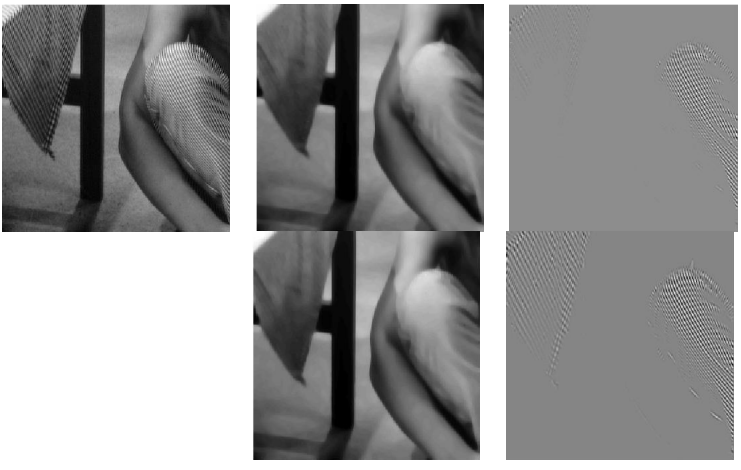


Fig. 1. Decomposition results (Up row: Starck's model; Bottom row: model (7))

Table 1. Comparison of SNR for two model (β=2.25, Unit: dB)

α	0.05	0.13	**0.24**	0.35	0.72	0.81	1.20
Starck's model	31.314	33.745	**35.367**	35.501	34.286	34.933.	32.049
Modified model	33.105	35.077	**37.423**	36.985	35.754	34.956	32.732

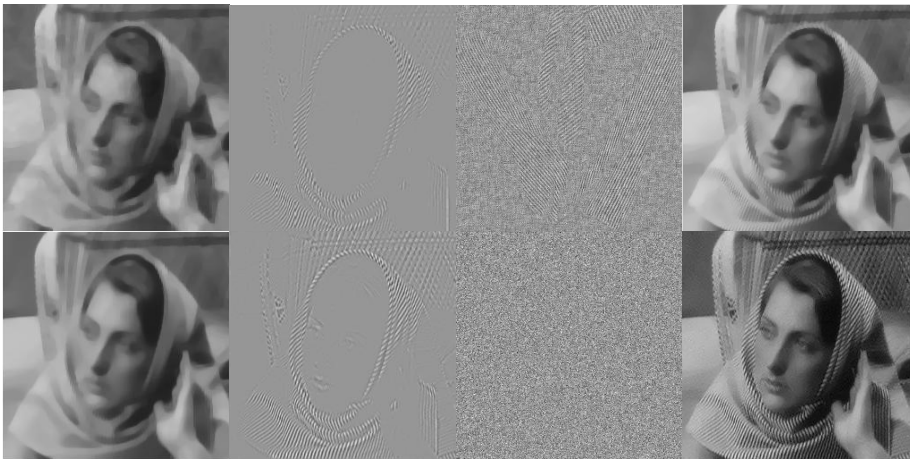


(a) Initial image (b) Noisy image (SNR=29.18dB)

Fig. 2. Test images

(2). Decomposition and Denoising for Noisy Image: In this experiments, we firstly add Gaussian noise with variation 35 for Woman image, and then use both Starck's model and modified model (9) to separate this noisy image into three components, structure component u, texture v and noise n. Through $u+v$, we obtain denoised image. As following Fig. 2 and Fig. 3, whereas, Starck's model can well separate

three components, it may lose some texture, for instance, some texture on scarf are taken as noise and merged into it. However, model (9) well overcome this drawback, and most small scale texture information are kept in texture component, while also well can keep edges in denoised image. In addition, we show SNR comparisons for both models when fixing $\beta = 2.25$ and varying α. As following Table 1, the improvement for SNR is the most obvious when set $\alpha = 0.24$ and use model (9) instead of Starck's model. So our results in Fig. 3 are obtained when set parameters $\alpha = 0.24$, $\beta = 2.25$.



(a) u (b) v+150 (c) n (d) u+v

Fig. 3. Results of decomposition and denoising (up row: Starck's model; bottom row: modified model (9))

5 Conclusion

Through studying of the last advance in CHA (calculate harmonic analysis) , such as second generation curvelet and wave atom, we combine Meyer's oscillatory pattern theory with these fruits to deeply research Starck's a novel image decomposition model, which easily cause losing of the small scale texture information. So we modify the Starck's model in two different cases, and have seen that our proposed model presents several advantages compared to Starck's model. Due to the second generation curvelets and wave can well represent structure and texture respectively. The algorithm well separate oscillating parts such as texture and noise from piecewise smooth parts. The total variational semi-norm is added for restricting structure parts, and generalized homogeneous Besov norm proposed by Meyer for restriction noisy components. Experiments show that the presented models are very robust to noise, and that can keep edges and textures stably.

Acknowledgments

I gratefully acknowledge partly funding by Chongqing University of Arts and Sciences, main project Z2008SJ13.

References

1. Meyer, Y.: Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. American Mathematical Society, Providence (2001)
2. Chen, S., Donoho, D., Saunder, M.: Atomic Decomposition by Basis Pursuit. SIAM Journal on Scientific Computing 20(1), 33–61 (1998)
3. Starck, J.L., Candes, E., Donoho, D.: Astronomical Image Representation by the Curvelet transform. Astronomy and Astrophysics 398(2), 785–800 (2003)
4. Aujol, J.F., Chambolle, A.: Dual Norms and Image Decomposition Models. International Journal of Computer Vision 63(1), 85–104 (2005)
5. Starck, J.L., Elad, M., Donoho, D.L.: Image Decomposition: Separation of Texture from Piecewise Smooth Content. In: Proceedings of SPIE, San Diego, California, USA, vol. 5207, pp. 571–582 (2003)
6. Aujol, J.F., Aubert, G., Blanc-Féraud, L., et al.: Image Decomposition Into a Bounded Variation Component and an Oscillating Component. Journal of Mathematical Imaging and Vision 22(1), 71–88 (2005)
7. Vese, L., Osher, S.: Modeling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing. Journal of Scientific Computing 19(1), 553–572 (2003)
8. Bai, J., Feng, X.: Image Decomposition by Curvelet Transform. Acta Electronics China 35(1), 123–126 (2007)
9. Osher, S., Sole, A., Vese, L.: Image Decomposition and Restoration Using Total Variation Minimization and the H-1 Norm. Journal of Multiscale Modeling and Simulation 1(3), 349–370 (2003)
10. Candes, E.J., Donoho, D.L.: Curvelets: a Surprisingly Effective Nonadaptive Representation for Objects with Edges, pp. 1–16. Department of Statistics, Stanford University, Stanford (1999)
11. Candes, E.J., Donoho, D.L.: New Tight Frames of Curvelets and Optimal Representations of Objects with Piecewise C2 Singularities. In: Applied and Computational Mathematics, California Institute of Technology, Pasadena, California, USA, pp. 1–68 (2002)
12. Candes, E.J., Demanet, L., Donoho, D.L.: Fast Discrete Curvelet Transforms. In: Applied and Computational Mathematics, California Institute of Technology, Pasadena, California, USA, pp. 1–43 (2005)
13. Demanet, L., Ying, L.X.: Wave Atoms and Sparsity of Oscillatory Patterns. Appl. Comput. Harmon. Anal. 23(3), 368–387 (2007)

Theoretical and Empirical Analysis of the Learning Rate and Momentum Factor in Neural Network Modeling for Stock Prediction

Jinchuan Ke[1], Xinzhe Liu[2], and Guan Wang[1]

[1] Beijing Jiaotong University, 100044, China
{jchke99,guanwang2005}@126.com
[2] Richest Investment Management, Inc. Ltd., 99501, USA
xzcindy99@yahoo.com

Abstract. Neural Network training requires a large number of learning epochs. An appropriate learning rate is important to the overall performance of the training. Under a weight-update algorithm, a low learning rate would make the network learning slowly, and a high learning rate would make the weights and error function diverge. To optimize the model parameters, this paper presents theoretical and empirical analysis of learning rate in neural network modeling for its application in stock price prediction, an increasing learning rate approach is suggested for practice. The effect of momentum factor is also investigated to speed up the convergence for network training.

Keywords: Neural Network, Learning rate, Momentum factor, Stock, Prediction.

1 Introduction

A Neural Network usually consists of a set of nodes and looks like a tree. The nodes are usually partitioned into different layers and fully connected together if two nodes are within neighboring layers. Each node can have incoming weight connections from the previous layer and outgoing weight connections to the next layer as shown in Figure 1.

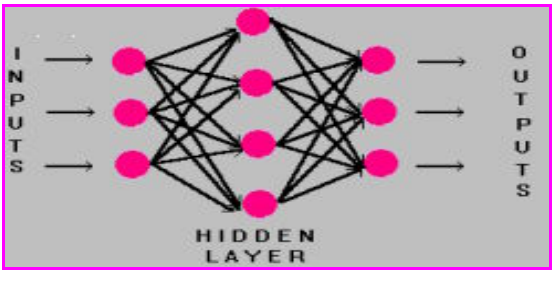


Fig. 1. Multi-Layer Perception (MLP) Network

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 697–706, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Each node in the network has a bias initialized from the input patterns and is usually normalized. During training each node obtains bias by adding up the outputs from the previous layer, multiplied by their weights respectively. By applying an activation function which is typically non-linear, the node's bias becomes a net input and is passed to the next layer. The introduction of an activation function is necessary to make the whole system capable of dealing with non-linear problems. After every node from a given layer has been evaluated, the nodes in the successive layers are evaluated until the final output nodes are reached [1]. During training, the above step is followed by a backward error-propagation step where all weights and biases are adjusted if there is a difference between the targeted and the actual output. The neural network training usually requires thousands of forward-backward weight-updated processes (epochs) and the system stops when the training output is close to the target output.

One of the common learning algorithms is "online Backpropagation", it is also called Vanilla algorithm, because it updates the weights after every training pattern. It is defined as follows [2]:

$$\Delta w_{ij} = \eta \delta_j o_i \tag{1}$$

$$\delta_j = \begin{cases} f_j^{'}(net_j)(d_j - o_j), & \textit{if unit j is an outputunit} \\ f_j^{'}(net_j)\sum_k \delta_k w_{jk}, & \textit{if unit j is a hiddenunit} \end{cases}$$

Where, η denotes the learning rate;

δ_j is the weighted sum of nodes k to which it has connections of w_{kj}.

$f^{'}(net_k)$ is the slope of the node nonlinearity at its current activation value.

O_i is the network actual output.

d_j is the desired output.

One method to speed up the learning is to introduce the curvature of the error surface which requires the computation of the second order derivatives of the error function. Quickprop is one of the algorithms. It assumes the error surface to be locally quadratic and attempts to jump in one step from the current position directly into the minimum of the parabola. Quickprop computes the derivatives in the direction of each weight [3]. A direct step to the error minimum is motivated after computing the first gradient with Back-propagation algorithm. The actual weight change is calculated as follows [4]:

$$\Delta(t+1)w_{ij} = \frac{S(t+1)}{S(t) - S(t+1)}\Delta(t)w_{ij} \tag{2}$$

where, w_{ij} is the weight between units i and j.

$\Delta(t+1)$ is the actual weight change.

$S(t+1)$ is the partial derivative of the error function by w_{ij}.

$S(t)$ is the last partial derivative.

Momentum algorithm introduces a momentum factor to speed up the learning and makes the new weight changed as:

$$\Delta w(t) = -\eta \frac{\partial E}{\partial w}(t) + \alpha \cdot \Delta w(t-1) \tag{3}$$

where, α is a constant specifying the influence of the momentum; E is the network error faction.

Momentum algorithm is the most common used and effective algorithm. From a case study of stock prediction, it can be observed from Figure 2 that the momentum algorithm provides the best training performance. This is due to the fact that the momentum term introduces the old weight change as a parameter for the computation of the new weight change. It avoids oscillation problems common with the regular Back-propagation algorithm.

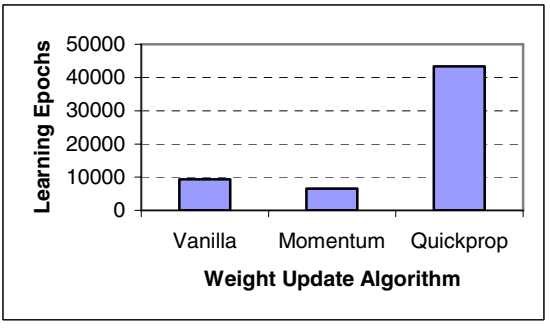


Fig. 2. Learning Epochs for various Algorithms

The test problem is based on the daily index quotes of TM (Toyota Motor Inc.) from website http://finance.yahoo.com/. In the network design, the Open price, S&P500 index, Relative strength index (RSI), and Stochastics (Raw-K) [5] are introduced as an input attribute (the input pattern optimization problem will be presented in my other investigations), and the BP standard net is applied for the network architecture.

The effect of the enhancements in Momentum algorithm is that the error surface traverses relatively rapidly with a few big steps, while the step size is decreased as the surface gets rougher [6]. This adaption of the step size increases learning speed significantly. It may be noted that the old weight change is lost every time the parameters are modified, new patterns are loaded, and the network is modified. Thus, the following study will be focused on the analysis of learning rate and momentum factor based on the momentum weight-updated algorithm.

2 Analysis of Learning Rate

In the training of a Neural Network, a low learning rate will make the training very slow. If the rate is too high, it will make the weights and the error function diverge. Based on the error function, the learning rate can be changed or fixed during the training process. Haykin suggested the following learning rate for quadratic error function [7]:

$$\eta = \frac{\eta_0}{\frac{n}{N/2} + \frac{c_1}{\max\left\{1, c_1 - \frac{\max\{0, c_1(n - c_2 N)\}}{(1-c_2)N}\right\}}} \tag{4}$$

where, η is the learning rate;

η_0 is the initial learning;

N is the total training epochs;

n is the current training epoch;

c_1, c_2 are constants.

If the error function has numerous local and global optima, as it is typical in feed-forward networks with hidden units, the optimal learning rate often changes dramatically during the training process.

With batch training, however, there is no need to use a constant learning rate. This is due to vastly efficient and convenient batch training algorithms (such as Quickprop and RPROP) that can be used to train the network. With incremental training, it is much more difficult to use an algorithm that automatically adjusts the learning rate. Various approaches have been suggested in the literatures, but most of them don't work in practice.

An attempt to train a neural network using a constant learning rate is usually a tedious process. It requires a number of trials. In the following sensitivity analysis, various constant learning rates will be examined followed by an analysis with increasing learning rates.

High-accuracy on training data requires a large number of learning epochs. For example, to obtain 90 percent accuracy in the training with network architecture of one hidden layer, 4 inputs and 500 samples, it has to run more than 2 million iterations and 50,000 epochs. Therefore, it is important to find an "optimum" learning rate to generate a smaller error which in turn would make the algorithm converge in less than millions of iterations. For example, let us set the stop criteria as:

(i) Average error < 0.0002

(ii) Epochs since minimum average error > 20,000

It can be seen from Figure 3 and Figure 4 that training situations with learning rate (η = 0.15, 0.20, 0.25, 0.30, 0.35) only meet the second stop criteria. Other cases with learning rate less than 0.15 meet the first stop criteria. The training process is very slow, and the training stops after the epochs reach a value of 20,000 since the minimum average error. With a learning rate of more than 0.35, as the learning rate is increased, the minimum average error tends to be higher when it reaches the second stop criteria. It is quite possible that the training will be diverge for the high learning rate although the performance is still very good in this example where the R-squared value is more than 0.9 for each case.

Similar results can be derived for the "low" or "high" learning rates which can be seen in Figure 5. In the case of learning rate η = 0.05, the epochs goes up to 97,622. The reason for higher epochs is that the training is too slow which needs large number of iterations for convergence, but it does not diverge. For learning rate as η = 0.5, the

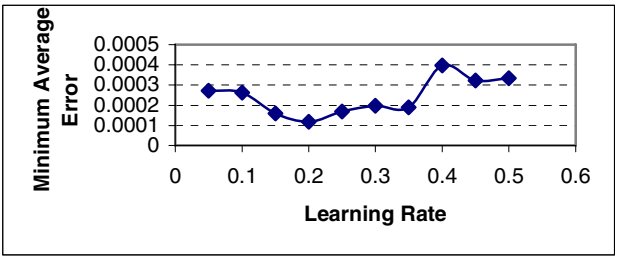


Fig. 3. Min. Ave. Error. as a Function of Learning Rate

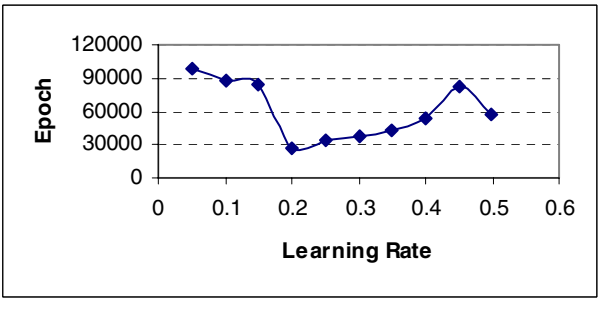


Fig. 4. Epoch as a Function of Learning Rate

training epoch is 56,773. The performance is fair compared to $\eta = 0.05$, but the minimum average error is as high as 0.000434. Improvement can not be made within the 20,000 epochs. The minimum error condition (less than 0.0002) is never achieved in this case.

Figure 6 is a plot of the RMS value as a function of the learning rate. Low and high learning rates result in a high RMS value which is greater than \$1.0. The accuracy is improved significantly when the learning rate is kept within the range between 0.15 and 0.35. This is indicated by a low RMS value of \$0.7. From this analysis, the range of optimum learning rate appears to be between 0.2 and 0.3 for the problem discussed here.

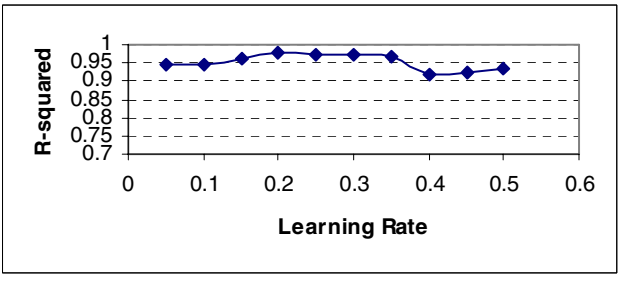


Fig. 5. R^2 as a Function of Learning Rate

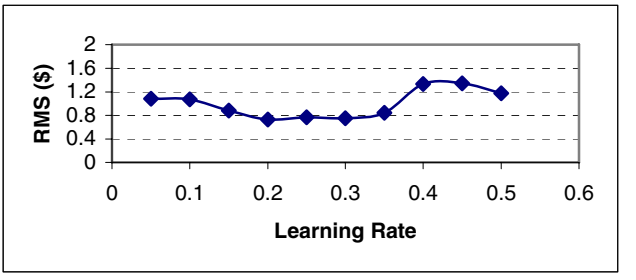


Fig. 6. RMS Value as a Function of Learning Rate

Weights in the network are updated after each pattern analysis, and all the nodes include a bias input which is part of the optimization process. The overall error decreases as the training proceeds. To improve the performance of training, one can apply an increasing learning rate by slightly adjusting the initial learning rate during the training. From the above discussion, it may be evident that for a good performance, the learning rate is between 0.2 and 0.3. One of the stop criteria is that the training stops after the epochs have reached 20,000. As a result, the increase in learning rate can be calculated as (0.35-0.15)/20,000 = 0.0001. For the purpose of sensitivity analysis, several simulation experiments are conducted by setting the initial learning rate as (0.2+0.3)/2 = 0.25, and then increasing the learning rate incrementally by an amount of ±0.00005.

Figure 7 is a plot of the minimum average error as a function of increasing learning rate. The stop criteria are set to be: Average error < 0.0002; or allowing the training to run upto 20,000 epochs since achieving the minimum average error. It can be noted that the training stops under the first criteria when the increasing learning rate is greater than (– 0.00002).

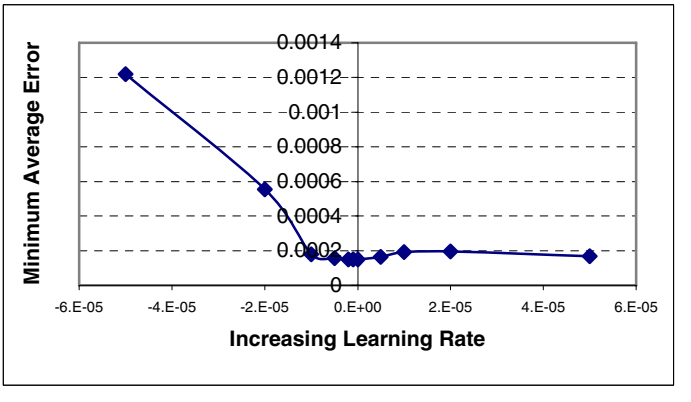


Fig. 7. Min. Ave. Error vs. Increase Learning Rate

Figure 8 is a plot of total learning epochs for each incremental in learning rate. The figure shows that the number of learning epochs does not change with the increasing

learning rate which varied from -0.000005 to 0.00001. With an increasing learning rate greater than 0.00001 or less than –0.000005, the learning epochs increase dramatically. Thus, it can be concluded that the "optimal" learning rate is 0.25 for the problem analyzed here.

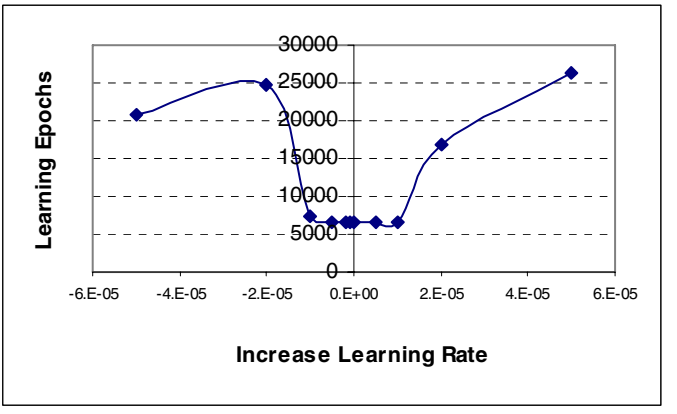


Fig. 8. Number of Epochs as a Function of Increase Learning Rate

3 Interaction of Learning Rate and Momentum

In momentum-mode update algorithm, the weight change is a combination of the previous weight change and the error gradient. The momentum factor is in the range of $0 < \alpha < 1$. Under the introduction of momentum factor α, the learning rate and momentum factor interact each other, leading to an accelerated learning. The iteration of momentum factor and learning rate can be illustrated by the expansion of equation (3) [6]:

$$\begin{aligned}\Delta W(t) &= -\eta \frac{\partial E}{\partial W}(t) + \alpha \cdot \Delta W(t-1) \\ &= -\eta \frac{\partial E}{\partial W}(t) + \alpha\left(-\eta \frac{\partial E}{\partial W}(t-1) + \alpha\left(-\eta \frac{\partial E}{\partial W}(t-2) + \ldots\right)\right) \\ &= -\eta \sum_{k=0}^{\infty} \alpha^k \frac{\partial E}{\partial W}(t-k)\end{aligned} \tag{5}$$

The weight update is now an exponential average of all the previous gradient terms. Because $0 < \alpha < 1$, the error from earlier derivative epochs decays with each time step. The time-constant of the system is controlled by α. For a small α, the coefficients decay quickly as k increases; For large $\alpha \rightarrow 1$, however, the coefficients decay very slowly. The system is stable but slow to react to changes in the error term. The momentum factor tends to amplify the effective learning rate and lead to faster convergence, when a long sequence of weight changes are all in the same direction.

However, the derivative $\partial E/\partial w(t)$ is usually not a constant, and the weight changes are affected by the past cycles, then the effect of the momentum is unknown. For the determination of the effective value with the learning rate, several cases are presented as follows:

Figure 9 shows the learning epochs as a function of momentum factor under the conditions of BP-Standard-Net architecture, random pattern selection, one hidden layer, with a learning rate of 0.1and minimum average error of 0.0002. The learning epochs decrease as the momentum increases. Without momentum factor, the learning epochs are about 17,000, with the momentum factor equals to 0.8, the training reaches the fastest convergence with the learning epochs of 4000. After the momentum factor of 0.8, the training needs more learn epochs.

Figure 10 shows the learning epochs as a function of momentum factor under the conditions of BP-Standard-Net architecture, random pattern selection, one hidden layer, with a learning rate of 0.3 and minimum average error of 0.005. The performance is similar to that from Figure 9. With the momentum factor equals to 0.8, the training reaches the fastest convergence. With a momentum factor greater than 0.8, the number of learning epochs for convergence increases. At the point $\alpha = 0.5$, the system becomes unstable with more epochs than that of $\alpha = 0.5$. However, the change is abrupt. For $\alpha > 0.5$, it recovers quickly. The reason for this abnormal problem might be caused by the "local minimum" or the noisy data.

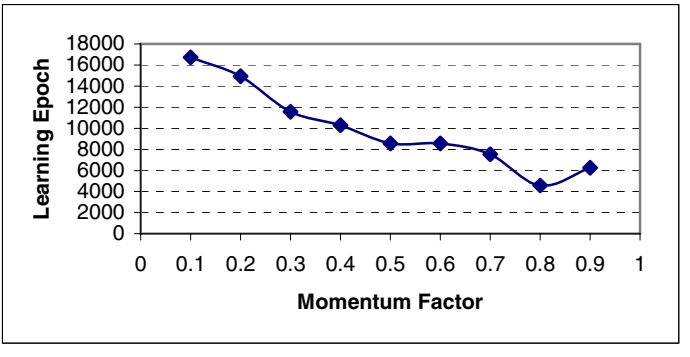


Fig. 9. Learning epoch as a function of momentum factor (with standard-net architecture, random pattern selection, one hidden layer, learning rate of 0.1, minimum average error of 0.0002)

The effective performance can also be obtained from combination of learning rate and momentum factor. To save pages, the other test cases are not presented here, however, the test results show the same fact that: (a) Under certain number of learning rate, convergence time decreases as the momentum factor increases. It smoothes the weight changes by filtering out high frequency variations. The add-up components damp out the side-to-side oscillations which caused by the consecutive opposing weight changes; (b) At the low end of the momentum range, the increase of momentum factor generally leads to faster convergence. The curves are smooth, unstable

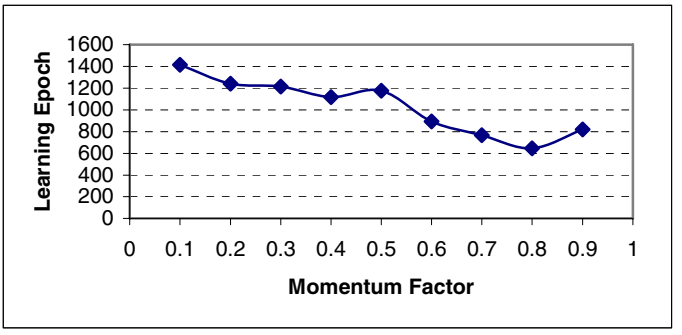


Fig. 10. Learning epoch as a function of momentum factor (with standard-net architecture, rotation pattern selection, one hidden layer learning rate of 0.3, minimum average error of 0.005)

abrupt error may occur, but it recovers quickly; (c) When the momentum is too large ($\alpha > 0.9$), the system performs poor, even diverse in some cases.

4 Conclusions

Neural Network training requires a large number of learning epochs. If the learning rate is too small, it makes the network learning very slow; if it is too high, it makes the weights and error function diverge. An appropriate learning rate is important to the overall performance of the training, although choosing a constant learning rate is usually a tedious process which requires a number of trials. From this study, it is found that the learning rate between 0.2 and 0.3 is suitable for the NN stock prediction. To improve the performance of training, one can apply an increasing learning rate by slightly adjusting the initial learning rate during the training.

Momentum may sometimes help the system speed up the learning process, and can also escape small local minima by providing a vector inertia. With momentum, weight changes are affected by error information from past cycles, the larger the momentum, the stronger influence on the previous changes. However, too much inertia will make the system sluggish. As the momentum factor closes to 1.0, the error function will be diverged. Therefore, a little inertia is useful for stabilization to smooth the weight changes by filtering out high frequency variations, but it can not totally eliminate the “local minimum”.

References

1. Nilsson, J. (ed.): Artificial Intelligence: A New Synthesis. Morgan Kaufmann Publishers, Inc., San Francisco (1998)
2. Pao, Y.-H. (ed.): Adaptive Pattern Recognition and Neural Networks. Addison-Wesley Publishing Company, Inc., Reading (1999)
3. Vidyasagar, M. (ed.): A Theory of Learning and Generalization. Springer, Heidelberg (1997)

4. NeuroShell2 User's Manual, Ward System Group, Inc., Frederick, MD (2002)
5. McNelis, P.D. (ed.): Neural networks in Finance: Gaining predictive edge in the market. Elsevier Academic Press, Amsterdam (2005)
6. Reed, R.D., Marks, R.J.: Neural Smithing, Massachusetts Institute of Technology (1999)
7. Haykin, S.: Neural networks: A Comprehensive Foundation. MacMillan, New York (1994)

Exploring Building Blocks through Crossover

Zhenhua Li[1,2] and Erik D. Goodman[3]

[1] Wuhan University, Wuhan 430072, China
[2] China University of Geosciences, Wuhan 430074, China
[3] Michigan State University, East Lansing, MI 48824, USA

Abstract. For binary-represented problems with strong building block structure and low epistasis, one-point crossover success (both offspring are better or one is better and the other not worse) and crossover failure (both offspring are worse or one worse and one not better) indicate that a building block was produced or broken near the crossover point. Starting from such a crossover point, and flipping bits ahead of and behind the crossover point until the fitness no longer decreases, gives an indication of the building block extent. The resulting string can be extracted and used to replace the corresponding part of the best individual found to date, to increase its fitness. Experiments on test functions Royal Road R1 and R2, Holland's Royal Road Challenge function and the H-IFF function show that such a method could improve performance significantly on the first 3 functions but be trapped on the last, relative to a classical GA.

1 Introduction

According to the Building Block (BB) Hypothesis [4,8], as confirmed by many theoretical studies [1,14], BBs play a fundamental role in genetic algorithm (GA) performance. Applications of BB theory remain diverse. Depending on how the BBs are obtained, existing research can be classified into two groups: constructing BBs or evolving BBs. The former explores BBs explicitly in an artificial way, while the latter exploits them only implicitly, through operations designed to preserve them, in a more natural way.

The first group uses special operations to find the BB, or linkage. One popular method is perturbation, a way to determine the relationship between genes by noting the fitness difference that results from a controlled change to one or more genes. For example, messy GA (mGA)[6], and its later improvements, fast mGA [5] and Gene Expression mGA [9], generate all substrings up to a certain length during the initialization procedure, then filter out BBs; LINC (linkage identification by nonlinearity check) and LIMD (linkage identification by nonmonotonicity detection) detect linkage by observing the fitness change resulting from perturbations in a pair of loci [12]. Another way to identify the BBs is based on mathematical analysis, to set up the relationship between genes by analyzing their mutual information. For example, Wyatt et al. discovers BBs from the Hessian matrix of the problem landscape [17].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 707–714, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The other group, following the classical GA process, adopts special representations, operators or breeding based on the BB hypothesis. For instance, the Linkage Learning GA (LLGA) employs a circular chromosome and an exchange operator [7]; BOA produces offspring based on a Bayesian network [13]; on our previous work, we design a learning crossover to avoid building block disruption [10].

Chen et al. proposed a classification of BB-based applications into three categories: perturbation, model building and chromosome and genetic operators [2]. Based on this, the first category described above and mathematical analysis method would be grouped, and the last two categories are combined since they have in common the evolution of BBs rather than artificial identification of them.

This paper presents a method to discover BBs, guided by crossover outcomes. When both offspring of a crossover are better than their parents (or one better and one not worse), it is taken to indicate that at least one BB of the parent includes the current recombination point. When a crossover fails, meaning that both offspring are worse than their parents (or one is worse and one not better), it is taken to indicate that at least one BB of the parent is disrupted by the current recombination point [10]. Working outward from such point, BBs can be experimentally determined by perturbation of neighboring loci. This approach falls into the first group.

This paper first explains why and what types of crossover outcomes could be an indication of a BB, then proposes a genetic algorithm to detect BBs based on successful and failed crossovers. It uses them in a simple way, with its effectiveness illustrated by results on four test functions. Finally, conclusions are drawn on Section 4.

2 Identifying Building Blocks Using Crossover

2.1 One-Point Crossover and Building Blocks

For those problems with strong building block structures and relatively low epistasis (interacting BBs are not widely distributed on the chromosome), the fitness difference after crossover may provide some information about BBs. The change of fitness of the offspring falls into one of six cases:

1. Both increase,
2. One increases and one does not change,
3. One increases and one decreases,
4. Neither changes,
5. One decreases and one does not change,
6. Both decrease.

If both offspring are no worse than their parents (cases 1 or 2) after one-point crossover, it means that building block(s) is/are formed near the crossover point; on the other hand, in cases 5 and 6, at least one building block is disrupted.

Figure 1 shows how one-point crossover involves in the creation and destroy of BBs.

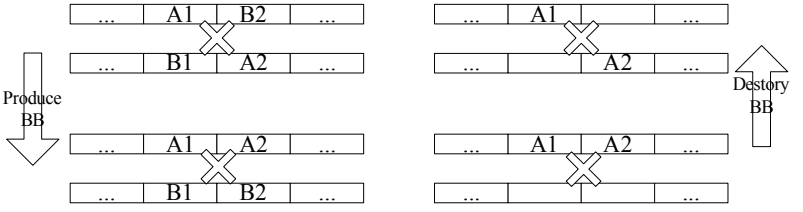


Fig. 1. One-point crossover producing or destroying BBs (left: two BBs involved, right: one BB involved)

2.2 Exploring Building Blocks from a Crossover Point

Some fitness changes indicate that a BB is produced or destroyed near the crossover point, so the BB could be detected by perturbations of genes near that point. Starting from the point, a BB can be found by flipping bits in each direction until the fitness no longer decreases.

Figure 2 illustrates how to find a building block through flipping bits one by one.

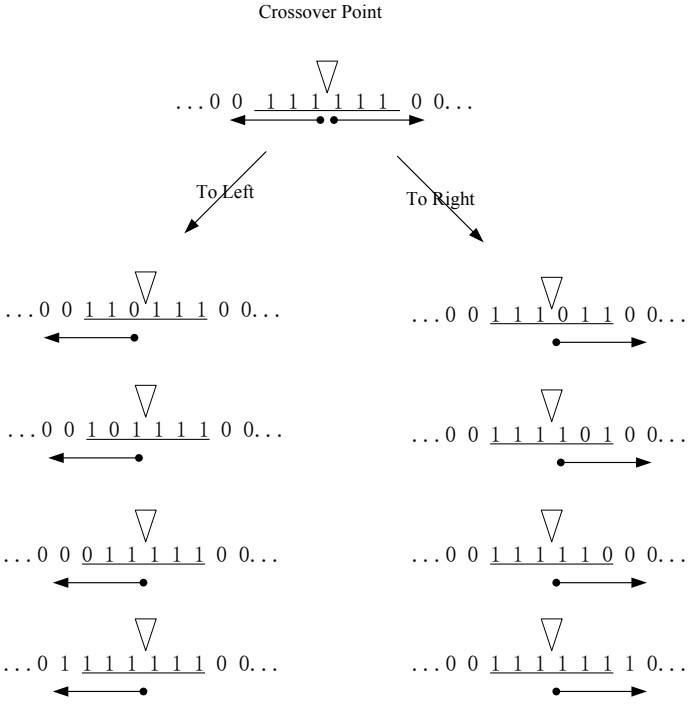


Fig. 2. Detecting a BB from the crossover point(here assuming 1's contribute more than 0's to fitness, and that the 1's originally shown constitute a BB)

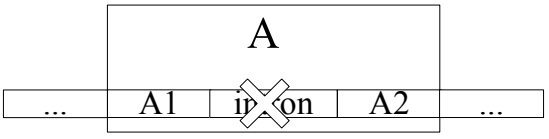


Fig. 3. Crossover points on intron(here A=A1+intron+A2)

However, crossover that takes place in the introns could also cause the offspring to be "better" or "worse." Under such a circumstance, building blocks on two sides of the intron may be combined to form a higher-level building block, but the process of flipping bits within the intron will yield no change and the process will stop without finding a BB (Fig.3).

2.3 Algorithms

Rather than using a traditional generational GA method of generating the next generation using serial application of operators (selection, crossover and mutation) to a large subset of the population, including possible crossover and subsequent mutation in production of any given offspring, we will perform only one (crossover or mutation) operation in generation of an offspring (or pair of offspring, through crossover). The evaluation of each new individual provides both its fitness and, in the case of crossover, information about the outcome for tracking crossover success or failure. Moreover, an additional operation-to test BBs in the context of the best individual found to date-is employed to find new elite individuals.

The algorithm is:

```
Initialize population;
Evaluate population;
WHILE not done
  CASE Operation OF
    Select for survival:
      Select one individual for survival unchanged to the next generation
    Crossover:
      Select two individual
      Cross them over
      Evaluate them
      Determine whether the crossover point resulted in either "failure"
      or "success"
        IF yes, record it and find the extent of the building
        block through flipping bits
        ELSE CONTINUE
    Mutation:
      Select one individual, mutate one bit
      Evaluate it
```

Replace the current population
Test the BBs in the context of the best-to-date individual, and
accept the changes yielding improved fitness
END WHILE

3 Experiments on R1, R2, RR(JH) and H-IFF

3.1 Test Functions

Progressively harder and harder, four test examples were chosen to examine the efficiency and adaptability of the proposed method.

The Royal Road Function R1 is designed to discover 64-bit "1"s from random binary strings[11,3]. Every continuous 8-bit substring, starting from any octuple locus, forms a building block, with 8 points awarded if all "1"s, and zero points otherwise. So the maximum fitness string is

"11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111"
and has a total fitness of 64 points. It can be expressed as
(B1,B2,B3,B4,B5,B6,B7,B8)
and its score is $B1 + ... + B8$. For example[3],
R1(1111111100...0) = 8, and R1(1111111100...011111111) = 16.

Royal Road Function R2 shares the same objective as R1, to find all "1" strings[11,3]. However, the fitness evaluation is hierarchical, with 4 levels, with the total score being the sum of all levels scores. It can be expressed as

(((B1, B2), (B3, B4)), ((B5, B6) , (B7, B8))) ,

and the higher-level BB within each pair of parenthesis gets a double bonus if the lower-level BBs are all formed. For example[3]:

R2(1111111100...011111111) = 16, but R2(111111111111111100 ... 0) = 32, and R2(11111111...1) = 192.

Holland's Royal Road Challenge, RR(JH), which was presented by John Holland at ICGA '93, then posted on the GA-List, was designed to test a GA's performance on BB discovery and combination. It is both hierarchical and deceptive. The total score sums the bonus from good combinations and the parts with correct schemata. In our experiments, we take the same parameter setting as Holland did, with a 240-bit string length, every 8-bit BB gapped with a 7-bit intron, and a negative reward (deception) for any 8-bit BB with more than 4 but fewer than 8 1's.

Hierarchical-if-and-only-if(H-IFF) is a well-known benchmark recursive function for defining hierarchically consistent decomposable fitness functions from two base functions, F(B) and T(B) [15,16]. T(B) evaluates the relationship of BBs, and F(B) gives the score according to both T(B) and the score of BBs in the next level. For example, for a four variable problem with blocks[15]:

F(a,b,c,d) = 4f(t(t(a,b),t(c,d))) + 2f(t(a,b)) + 2f(t(c,d)) + f(a) + f(b) + f(c) + f(d).

Therefore, the higher level score could be multiplied only if its lower blocks are consistent. For instance, though "00000" and "1111" are good building blocks alone, but its combination "000001111" gets no bonus. The Hamming Distance between the second global optimum and the global optimum is longer than any other pair of optimums, and it is same for the reward difference. Such a feature make it is easy to be trapped in local optima, since it is hard to be happen to replace the whole inconsistent part accurately to achieve the best result. It could be named as combination deception or structure deception, comparing to so-called well-known building block deception. Here, a 64-bit problem is studied.

3.2 GA Setting

We use the same settings for all of the above four test functions, except the population size (Tab.1).

Table 1. The GA Settings

	R1	R2	RR(JH)	HIFF
Crossover	one-point crossover			
Mutation	single-bit mutation			
Selection	tournament, size 4			
Pop. size	2000		5000	
P Selection	0.2			
P Crossover	0.7			
P Mutation	0.1			
No. of Gens.	50			
No. of Runs	100			

3.3 Results

Every example was run 100 times indepently, and their results are shown in Tab.2.

The method works on R1 and R2 about 6 times faster than normal ones. It only needs about 5000 evaluations. Considering that population size is 2000, the data shows that in most runs the global optima are achieved after 2-3 generations. That is to say, the BBs are identified quickly and mixed efficiently. Morever, R2 needs less evaluations than R1 both in GA and BBDGA, demonstrates that the royal road really works.

The result on RR(JH) is also positive, though it is a challenge for most GAs. It shows that the deceptive BBs are well discovered and then mixed properly.

However, it is easy to get trapped on H-IFF(lower success rate), though it has the less evaluations too. We guess that the method presented here has a stronger bias on exploit than explore, therefore it gets a fast convergence but take the risk to lost change to achieve the global optimum, on those stronger "combination deception" problems.

Table 2. Average Number of Evaluations of non-BB-detection GA and GA with BB-detection (BBDGA) to find global optimum on 50 generations(100 runs)

	GA Type	Success Rate	Mean	Std Dev	T-test
R1	GA	100%	30925	2411	45.8
	BBDGA	100%	5896	1287	
R2	GA	100%	29731	2971	36.2
	BBDGA	100%	5756	1450	
RR(JH)	GA	100%	107312	10016	36.6
	BBDGA	100%	25809	4856	
H-IFF	GA	95%	77593	9628	0.7
	BBDGA	79%	70561	44364	

4 Conclusions

For problems with strong building block structure and relatively low epistasis, one-point crossover can be used to reveal the existence of building blocks-certain crossover results can reveal whether the crossover point is within or outside a BB. Further mutation-based perturbation from the recombination point can detect the extent of the BBs thus discovered. Its performance on four test functions illustrates that it can improve on the efficiency of classical GA methods, on those strong building block structure, relatively low epistasis and less "combination deceptive" problems.

Acknowledgement

We wish to acknowledge the help of Prof. Lishan Kang and the support of State Key Laboratory of Geological Processes and Mineral Resource (No. GPMR200715) and the Michigan State University High Performance Computing Center.

References

1. Aporntewan, C., Chongstitvatana, P.: A quantitative approach for validating the building-block hypothesis. In: 2005 IEEE Congress on Evolutionary Computation 2005, IEEE CEC 2005, pp. 1403–1409 (September 2005)
2. Chen, Y.-P., Goldberg, D.E.: Tightness time for the linkage learning genetic algorithm. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 837–849. Springer, Heidelberg (2003)
3. Forrest, S., Mitchell, M.: Relative building-block fitness and the building-block hypothesis. In: Whitley, L.D. (ed.) Foundations of Genetic Algorithms 2, pp. 109–126. Morgan Kaufmann, San Mateo (1993)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)

5. Goldberg, D.E., Deb, K., Kargupta, H., Harik, G.: Rapid accurate optimization of difficult problems using fast messy genetic algorithms. In: Forrest, S. (ed.) Proc. of the Fifth Int. Conf. on Genetic Algorithms, pp. 56–64. Morgan Kaufmann, San Mateo (1993)
6. Goldberg, D.E., Korb, B., Deb, K.: Messy genetic algorithms: Motivation, analysis, and first results. Complex Systems 3(5), 493–530 (1989)
7. Harik, G.: Learning linkage to efficiently solve problems of bounded difficulty using genetic algorithms. PhD thesis, University of Michigan (1997)
8. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
9. Kargupta, H.: The gene expression messy genetic algorithm. In: International Conference on Evolutionary Computation, pp. 631–636 (1996)
10. Li, Z., Goodman, E.D.: Learning building block structure from crossover failure. In: Thierens, D., Beyer, H.-G., Bongard, J., Branke, J., Clark, J.A., Cliff, D., Congdon, C.B., Deb, K., Doerr, B., Kovacs, T., Kumar, S., Miller, J.F., Moore, J., Neumann, F., Pelikan, M., Poli, R., Sastry, K., Stanley, K.O., Stutzle, T., Watson, R.A., Wegener, I. (eds.) GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation, London, July 7-11, 2007, vol. 2, pp. 1280–1287. ACM Press, New York (2007)
11. Mitchell, M., Forrest, S., Holland, J.H.: The royal road for genetic algorithms: Fitness landscapes and GA performance. In: Varela, F.J., Bourgine, P. (eds.) Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 1991, Paris, pp. 245–254. A Bradford book, The MIT Press (1991)
12. Munetomo, M., Goldberg, D.E.: Identifying linkage groups by nonlinearity/nonmonotonicity detection. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, Orlando, Florida, USA, vol. 1, pp. 433–440. Morgan Kaufmann, San Francisco (1999)
13. Pelikan, M., Goldberg, D.E., Cantú-Paz, E.: BOA: The Bayesian optimization algorithm. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference GECCO 1999, Orlando, FL, vol. I, pp. 525–532. Morgan Kaufmann Publishers, San Francisco (1999)
14. Thierens, D., Goldberg, D.E.: Mixing in genetic algorithms. In: Proceedings of the Fifth International Conference on Genetic Algorithms, ICGA 1993, June 1993, pp. 38–45 (1993)
15. Watson, R.A., Hornby, G., Pollack, J.B.: Modeling building-block interdependency. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 97–106. Springer, Heidelberg (1998)
16. Watson, R.A., Pollack, J.B.: Hierarchically consistent test problems for genetic algorithms: Summary and additional results. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999), pp. 292–297. Morgan Kaufmann, San Francisco (1999)
17. Wyatt, D., Lipson, H.: Finding building blocks through eigenstructure adaptation. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1518–1529. Springer, Heidelberg (2003)

A Chaotic Neural Network Combined Heuristic Strategy for Multidimensional Knapsack Problem

Ying Zhou, Zhanghui Kuang, and Jiahai Wang

Department of Computer Science, Sun Yat-sen University, No. 135,
Xingang West Road, Guangzhou 510275, P.R. China
zhouy3@mail2.sysu.edu.cn,
wangjiah@mail.sysu.edu.cn

Abstract. Multidimensional Knapsack Problem (MKP), as a classic combinatorial optimization problem, is used widely in various fields such as capital budgeting, allocating processors and databases in a distributed computer system. In this paper, a chaotic neural network combined heuristic strategy (TCNN-HS) is proposed for MKP. The proposed algorithm combines heuristic strategy which includes repair operator and improvement operator so that not only the infeasible solution can be overcome, but also the quality of the solutions can be improved. The TCNN-HS is tested on some benchmark problems, which is selected from OR library. Simulation results show that the proposed approach can find optimal solutions for some instances and outperforms TCNN.

Keywords: Multidimensional Knapsack Problem, Chaotic Neural Network, Heuristic Strategy.

1 Introduction

The multidimensional knapsack problem (MKP) is a combinatorial optimization problem. It derives its name from the following maximization problem of the best choice of essentials that can fit into one bag to be carried on a trip. Many practical problems can be formulated as MKP, such as the capital budgeting problem, allocating processors and databases in a distributed computer system, project selection and cargo loading, and cutting stock problem [1]. MKP is also a NP-hard problem, which means that no algorithm exits that exactly solves the problem in polynomial time. Thus, it is important to develop fast algorithms that can find good approximate solutions. There are some heuristic approaches for solving this problem in the literatures, for examples, Drexl *et al* [2] suggested a simulated annealing approach to MKP, Reinaldo *et al* [3] proposed a Meta-RaPS approach for MKP, Shi [4] suggested an improved ant colony algorithm (ACO) and Lee [5] also proposed an approach using ACO for MKP.

Artificial neural networks (ANN) have turned out to be powerful in finding good approximate solutions to difficult combinatorial optimization problems [6]. Chen *et al* [7] proposed a method called transiently chaotic neural network (TCNN). It is proved to be more efficient and can prevent the network from being trapped in commonplace local minima, and may obtain acceptably near-optimal solutions eventually. Wang [8]

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 715–722, 2008.
© Springer-Verlag Berlin Heidelberg 2008

used TCNN model and rewrite the energy function using a method called Mean Field Approach (MFA) [6] for MKP. However, TCNN also has some disadvantage, for example, it cannot obtain the feasible solution for MKP all the time, and the solution strongly depends on initialization. In this paper, we make some modifications on TCNN model suggested by [8]. Furthermore, A simple heuristic strategy, which includes repair operator and improving operator, is introduced to the approach. The simulation indicates that TCNN-HS has a better performance than TCNN.

The remaining sections of this paper are organized as follows. Section 2 defines the MKP. The TCNN-HS for MKP is proposed in section 3. In Section 4, benchmark data sets are used to evaluate the proposed algorithm. Finally, some conclusions are drawn in last section.

2 Formulation of MKP

MKP can be described as follow [1] [6]: Given a set of n items and m knapsack. Let p_j, w_j and c_i be the profit of item j, the weight of item j and the capacity of the knapsack i, respectively. Note that p_j, w_j and c_i are positive real number.

Pick out the subset of items so as to maximize the utility,

$$f(x) = \sum_{j=1}^{n} p_j x_j \tag{1}$$

Subject to

$$\sum_{j=1}^{n} w_{ij} x_j \le c_i, \ i = 1,...,m \tag{2}$$

$$x_j \in \{0,1\}, \quad j = 1,...,n$$

where $x = (x_1, x_2, ..., x_n)$ is regarded as the item set in the knapsacks, x_i is set as follows:

$$x_i = \begin{cases} 1, \text{ the } i\text{th item in every knapsack} \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

3 TCNN Combined Heuristic Strategy for MKP

The transiently chaotic neural network [7] can be described as follows:

$$x_i(t) = \frac{1}{1 + e^{-y_i(t)/\varepsilon}} \tag{4}$$

$$y_i(t+1) = ky_i(t) + \alpha(\sum_{j=1, j\neq i}^{n} w_{ij}x_j(t) + I_i) - z_i(t)(x_i(t) - I_0) \tag{5}$$

$$z_i(t+1) = (1-\beta)z_i(t) \tag{6}$$

Where

$x_i(t)$ = the output of neuron i at discrete time t

$y_i(t)$ = the input of neuron i at discrete time t

$w_{ij} = w_{ji}; w_{ii} = 0; \sum_{j=1, j\neq i}^{n} w_{ij}x_j + I_i = -\frac{\partial E}{\partial x_i}$

E = energy function

$z_i(t)$ = self - feedback connection weight or refractory strength $(z_i(t) \geq 0)$

β = damping factor of the time - dependent $z_i(t)(0 \leq \beta \leq 1)$

I_0 = positive parameter.

We also use MFA [6] [8] [9] to rewrite the energy function for MKP as follow:

$$E = -\sum_{j=1}^{n} p_j x_j + \lambda \sum_{i=1}^{m} \Phi(\sum_{j=1}^{n} w_{ij}x_j - c_i) \tag{7}$$

where $\Phi(x)$ is given by

$$\Phi(x) = \begin{cases} x, \text{if } x > 0 \\ 0, \text{otherwise} \end{cases} \tag{8}$$

In order to avoid self-couplings altogether, References [6] [8] [9] suggest a method by replacing $\frac{\partial E}{\partial x_i}$ with the difference in E computation at $x_i = 1$ and $x_i = 0$ respectively. Therefore we have

$$-\frac{\partial E}{\partial x_i} \rightarrow p_i - \lambda \sum_{k=1}^{m} [\Phi(\sum_{j=1}^{n} w_{kj}x_j - c_k)|_{x_i=1} - \Phi(\sum_{j=1}^{n} w_{kj}x_j - c_k)|_{x_i=0}] \tag{9}$$

In general, the parameter λ for some problems can not be chosen easily since the value p_i is always much larger than the penalty factor. Therefore we make some modifications on p_i. The modification rule can be defined as follow:

$$p_i' = \frac{p_i - p_{\min}}{p_{\max} - p_{\min}} \tag{10}$$

where

$$p_{\max} = \max\{p_1, ..., p_n\}$$

$$p_{\min} = \min\{p_1, ..., p_n\}$$

Then p_i can be replaced with p_i' and the parameter λ can be set as 1 simply. The TCNN parameter I_0 can also be replaced with p_i' proposed by Xu [11]. Xu has already proved that I_0 played an important role in the dynamic equation. When the bifurcations disappear, the output of a neuron is close to I_0. Therefore, I_0 can be replaced with p_i' so that the item which has a higher profit can have more chances to be selected. During the process of the network running, at every iteration, we make the continuous output x_i discrete to get a solution (note that we still use the continuous output of each neuron to generate input of each neuron). The discrete rule can be described: at the end of each iteration, we just simply compute the average of all the continuous output of neurons, and compare it with each output. If the output is larger or equal to the average, it will be set as 1 otherwise 0. it can be formulated as follow

$$x_i' = \begin{cases} 1, \text{if } x_i \geq \text{average} \\ 0, \text{otherwise} \end{cases} \tag{10}$$

Then we use the discrete output to determine whether we obtain a feasible solution or better solution. As we know, TCNN can not get a feasible solution every time, and solution strongly depends on the initialization, therefore here we use a heurisitic strategy suggested by reference [10] to get a feasible, better solution. For the infeasible solution, we change it into feasible solution following the steps below:

Algorithm 1 the steps of the repair operator

Step 1. Calculate the profit density $\delta_{ij} = c_j \cdot p_i / w_{ij}$ for every item in every knapsack.

Step 2. Compute the lowest value of the profit density $\delta_i = \min\{c_j \cdot p_i / w_{ij}\}$ for every item.

Step 3. Sort and relabel items according to the ascending order or δ_i.

Step 4. Remove the corresponding items with lowest values of δ_i from the item set (i.e. change corresponding output 1 into 0).

Step 5. Repeat Step 4 until a feasible solution is found.

In order to get a better solution, we use improvement operator as follow:

Algorithm 2 the steps of the improvement operator

Step 1. Calculate the profit density $\delta_{ij} = c_j \cdot p_i / w_{ij}$ for every item in every knapsack.

Step 2. Compute the lowest value of the profit density $\delta_i = \min\{c_j \cdot p_i / w_{ij}\}$ for every item.

Step 3. Sort and relabel items according to the descending order or δ_i.

Step 4. Add the corresponding item with the highest values of δ_i. into the item set (i. e. change corresponding output 0 into 1).

Step 5. If one of knapsack constraints is not satisfied, then undo this modification and return to Step 4. Otherwise return to Step 4.

Thus The TCNN-HS algorithm process can be described as follow:

Algorithm 3 the algorithm of the TCNN-HS

Step 1. Set t=0 ,bestSolution=0 and initialize each $y_i(t)$ randomly in [-1,1].

Step 2. Compute each $x_i(t)$ using the Eq. (4).

Step 3. Compute each $y_i(t+1)$ using the Eq. (5) and (9).

Step 4. Make the output $x_i(t)$ discrete using the Eq. (10) and get a solution.

Step 5. If the solution is infeasible, then use the repair operator (Algorithm 1) to get a feasible solution.

Step 6. Use the improvement operator (Algorithm 2) to get a better solution.

Step 7. If bestSolution<currentSolution, then bestSolution=currentSolution.

Step 8. Set *t=t+1*. If *t* is equal to a given bound, then goto Step 9, else goto Step 2.

Step 9. Print the bestSolution and exit the procedure.

4 Simulation Results

To demonstrate the performance of TCNN-HS, it is tested on benchmarks of MKP selected from OR-Library (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/) on a notebook computer (Core 2 T7250 2.0GHz 1.99Ghz, RAM 1G). The test instances are WEING1-WEING8 and WEISH01-WEISH30.These instances, which the number of constraints and the number of items are from 2 to 5 and from 28 to 105, respectively, are well known with optimal values. Table 1 shows the detail features of the instances.

Table 1. The detail features of test instances

Problem	Size(m/n)	Optimal	Problem	Size(m/n)	Optimal
WEING1	2/28	141278	**WEISH12**	5/50	6339
WEING2	2/28	130883	**WEISH13**	5/50	6159
WEING3	2/28	95677	**WEISH14**	5/60	6954
WEING4	2/28	119337	**WEISH15**	5/60	7486
WEING5	2/28	98796	**WEISH16**	5/60	7289
WEING6	2/28	130623	**WEISH17**	5/60	8633
WEING7	2/105	1095445	**WEISH18**	5/70	9580
WEING8	2/105	624319	**WEISH19**	5/70	7698
WEISH01	5/30	4554	**WEISH20**	5/70	9450
WEISH02	5/30	4536	**WEISH21**	5/70	9074
WEISH03	5/30	4115	**WEISH22**	5/80	8947
WEISH04	5/30	4561	**WEISH23**	5/80	8344
WEISH05	5/30	4514	**WEISH24**	5/80	10220
WEISH06	5/40	5557	**WEISH25**	5/80	9939
WEISH07	5/40	5567	**WEISH26**	5/90	9584
WEISH08	5/40	5605	**WEISH27**	5/90	9819
WEISH09	5/40	5246	**WEISH28**	5/90	9492
WEISH10	5/50	6339	**WEISH29**	5/90	9410
WEISH11	5/50	5643	**WEISH30**	5/90	11191

Table 2. Simulation results of the TCNN and the proposed TCNN-HS

Problem	TCNN	TCNN-HS
	Best/Worst/Ave./Sd.	Best/Worst/Ave./Sd.
WEING1	141168/134023/139516.2/1754.3	**141278/**141258/141265.6/9.7
WEING2	130773/124714/129390.3/1368.8	**130883/130883/130883.0/0**
WEING3	**95677/**94517/95504.3/218.6	**95677/**95627/95675.0/9.8
WEING4	115501/114166/114809.2/333.8	**119337/119337/119337.0/0**
WEING5	98686/94517/96337.8/1205.2	**98796/98796/98796.0/0**
WEING6	**130623/**128358/129835.0/572.2	**130623/130623/130623.0/0**
WEING7	1090634/1063245/1078690.7/5737.5	**1095445/**1094662/1095316.7/142.4
WEING8	623137/558908/603215.5/16645.2	**624319/**621086/623547.4/824.5
WEISH01	**4554/**4386/4515.3/48.3	**4554/4554/4554.0/0**
WEISH02	**4536/**4345/4484.7/36.9	**4536/4536/4536.0/0**
WEISH03	**4115/**3910/4018.1/46.6	**4115/**4106/4108.3/3.9
WEISH04	**4561/**4468/4526.5/27.3	**4561/4561/4561.0/0**
WEISH05	**4514/**4358/4484.6/42.4	**4514/4514/4514.0/0**
WEISH06	**5557/**5293/5473.5/57.7	**5557/**5544/5545.0/3.5
WEISH07	**5567/**5303/5460.9/64.2	**5567/5567/5567.0/0**
WEISH08	**5605/**5256/5505.9/80.7	**5605/**5603/5604.8/0.5
WEISH09	**5246/**4940/5194.6/55.5	**5246/5246/5246.0/0**
WEISH10	6338/6058/6225.0/53.3	**6339/6339/6339.0/0**
WEISH11	**5643/**4788/5291.7/225.3	**5643/**5624/5642.2/3.7
WEISH12	6338/5705/6179.4/119.8	**6339/6339/6339.0/0**
WEISH13	**6159/**5624/5962.8/110.8	**6159/**6083/6132.6/32.7
WEISH14	**6954/**6455/6817.6/88.0	**6954/6954/6954.0/0**
WEISH15	**7486/**7096/7372.9/90.8	**7486/7486/7486.0/0**
WEISH16	**7289/**6980/7200.7/72.2	**7289/**7247/7281.3/14.0
WEISH17	**8633/**8418/8540.0/50.5	**8633/**8621/8626.4/5.9
WEISH18	9540/9213/9413.7/85.3	**9580/**9521/9535.5/17.9
WEISH19	7683/7123/7495.8/134.5	**7698/7698/7698.0/0**
WEISH20	9445/9037/9311.8/85.0	**9450/9450/9450.0/0**
WEISH21	**9074/**8728/8933.3/90.4	**9074/9074/9074.0/0**
WEISH22	**8947/**8193/8798.8/167.4	**8947/**8886/8931.7/21.6
WEISH23	8280/7704/8122.8/134.6	**8344/**8341/8341.1/0.6
WEISH24	10189/9667/10059.2/95.2	**10220/**10107/10157.4/32.0
WEISH25	9908/9562/9798.4/75.6	**9939/**9883/9902.4/15.0
WEISH26	**9584/**9013/9449.1/111.0	**9584/**9542/9579.7/7.8
WEISH27	**9819/**9325/9688.8/123.2	**9819/**9816/9816.7/1.3
WEISH28	9438/8795/9303.9/147.5	**9492/9492/9492.0/0**
WEISH29	**9410/**8661/9172.0/167.6	**9410/**9369/9373.5/5.3
WEISH30	11172/10771/11088.8/68.7	**11191/**11143/11165.7/11.1

The parameter α is relatives with convergence speed. If α is too large, then the network will convergence quickly, perhaps leading to miss some good solutions. But if it is too small, then the network will convergence too slowly. In the simulation

experiments, α is set to be 0.001 to balance the convergence speed and quality of solutions. The parameter β acts an important role in the chaotic annealing. The chaotic phenomenon will disappear quickly when β is too large while chaos is cooling very slowly when β is too small. Here, we set $\beta = 0.01$. The parameter $\varepsilon = 0.004, k = 0.9$, which adopt from reference [7] [8] directly.

In order to obtain a faster convergence, we use a method introduced by reference [8], by making ε annealing gradually. In this simulation, it is defined as follow:

$$\varepsilon(t+1) = 0.8 * \varepsilon(t) .$$

We run the TCNN-HS and TCNN [8] on each instance 50 runs, and calculate the best value, the worst value, the average value and SD for each instance. Table 2 shows the comparison between the TCNN-HS and TCNN. From the result table, we can see that TCNN-HS can find the exactly optimal solutions for each instance in 50 times and can find a good approximate solution (or optimal solution) every time for each instance. Although TCNN can find optimal solutions for few instances, it cannot obtain a good approximate solution every time. It is worth to notice that TCNN gets unexpectedly the worse results on some instance. Note that all the optimal obtained are in bold. Thus we can conclude that TCNN-HS outperforms TCNN and always acquire the better results than TCNN.

Table 3. The computation time of TCNN-HS

Problem	Time(sec)	Problem	Time(sec)
WEING1	0.05	**WEISH12**	0.29
WEING2	0.05	**WEISH13**	0.29
WEING3	0.05	**WEISH14**	0.41
WEING4	0.05	**WEISH15**	0.36
WEING5	0.05	**WEISH16**	0.38
WEING6	0.05	**WEISH17**	0.39
WEING7	0.51	**WEISH18**	0.47
WEING8	0.47	**WEISH19**	0.55
WEISH01	0.11	**WEISH20**	0.50
WEISH02	0.11	**WEISH21**	0.49
WEISH03	0.12	**WEISH22**	0.70
WEISH04	0.11	**WEISH23**	0.70
WEISH05	0.11	**WEISH24**	0.60
WEISH06	0.17	**WEISH25**	0.63
WEISH07	0.17	**WEISH26**	0.87
WEISH08	0.17	**WEISH27**	0.87
WEISH09	0.19	**WEISH28**	0.86
WEISH10	0.29	**WEISH29**	0.87
WEISH11	0.30	**WEISH30**	0.79

The computation complexity of TCNN-HS is $O(MN^2)$ in each iteration and it is quite close to the TCNN. Table 3 shows the computation time for TCNN-HS for each instance. We can find that TCNN-HS can get a good solution within a reasonable time

5 Conclusions

In this paper, a chaotic neural network combined heuristic strategy is proposed for MKP. In order to get a better, feasible solution, a heuristic strategy, which includes repair operator and improvment operator, is introduced to modify the infeasible solution and improve the quality of the solution. The simulation results show that TCNN-HS outperforms TCNN for MKP.

Acknowledgments. The research work described in this paper was supported by the Guangdong Provincial Natural Science Foundation of China (07300630), the Specialized Research Fund for the Doctoral Program of Higher Education (20070558052), and the Scientific Research Foundation for the Returned Overseas Chinese Scholars State Education Ministry.

References

1. Kong, M., Tian, P., Kao, Y.C.: A New Ant Colony Optimization Algorithm for the Multidimensional Knapsack Problem. Computers and Operations Research (2007)
2. Drexl, A.: Darmstadt: A Simulated Annealing Approach to the Muticonstraint Zero-One Knapsack Problem. Computing, 1–8 (1998)
3. Reinaldo, J.M., Gail, W.D., Gray, E.W.: Meta-RaPS Approach for the 0-1 Multidimenional Knapsack Problem. Computers & Industrial Engineering, 83–96 (2005)
4. Shi, H.X.: Solution to 0/1 Knapsack Problem Based on Improved Ant Colony Algorithm. In: 2006 IEEE International Conference on Information Acquisition, pp. 20–23 (2006)
5. Lee, C.Y., Lee, Z.J.: A New Approach for Solving 0/1 Knapsack Problem. In: IEEE International Conference on Systems, Man, and Cybernetics, October 8-11 (2006)
6. Mattias, O., Pi, H.: A Study of the Mean Field Approach to Knapsack Problems. Neural Networks 10, 263–271 (1997)
7. Chen, L.N., Aihara, K.: Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos. Neural Networks 8, 915–930 (1995)
8. Wang, B.Y., Dong, H., He, Z.Y.: A Chaotic Annealing Neural Network with Gain Sharpening and Its Application to the 0/1 Knapsack Problem. Neural Processing Letters, 243–247 (1999)
9. Ohlsson, M., Peterson, C., Söderberg, B.: Neural Networks for Optimization Problems with Inequality Constraints – the Knapsack Problem. Neural Computation. LU TP 92-11 (March 1992)
10. Li, H., Jiao, Y.C., Zhang, L., Gu, Z.W.: Genetic Algorithm Based on the Orthogonal Design for Multidimensional Knapsack Problems. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4221, pp. 696–705. Springer, Heidelberg (2006)
11. Xu, X.S., Tang, Z., Wang, J.H.: An Improved Transiently Chaotic Neural Network for the Maximum Independent Set Problem. International Journal of Neural Systems 14, 381–392 (2004)

A New Optimization Algorithm for Weight Optimization

Hui Li and Xuesong Yan

School of Computer Science, Chin University of Geosciences, Wuhan, 430074
huili@vip.sina.com

Abstract. Particle Swarm Optimization algorithm was developed under the inspiration of behavior laws of bird flocks, fish schools and human communities. Aiming at the disadvantages of Particle Swarm Optimization algorithm like being trapped easily into a local optimum, this paper improves the standard PSO and proposes a new algorithm to solve the overcomes of the standard PSO. The new algorithm keeps not only the fast convergence speed characteristic of PSO, but effectively improves the capability of overall searching as well. We use the new algorithm for the weight optimization in college student evaluation, and compared with PSO, the results show that the new algorithm is efficient.

Keywords: Particle swarm optimization, genetic algorithm, college student evaluation.

1 Introduction

Particle Swarm Optimization (PSO) [1] was an algorithm first presented in 1995 by Eberhart and Kennedy, and it was developed under the inspiration of behavior laws of bird flocks, fish schools and human communities. If we compare PSO with Genetic Algorithms (GAs), we may find that they are all maneuvered on the basis of flocks and schools. But PSO doesn't rely on genetic operators like alternative operators, cross operators or abnormal operators to maneuver individual particles; it optimizes collectively information mutuality among individual particles in any flocks or schools. PSO achieves its optimum solution by starting from a group of random solution and then searching repeatedly. Once PSO was presented, it invited widespread concerns among scholars in the optimization sphere and shortly afterwards it had become a studying focus within only several years. A number of scientific achievements had thus far emerged in the field [2] [3] [4]. PSO was proved to be a sort of high efficient optimization algorithm by numerous research and experiments [5]. This paper improves the disadvantages of PSO's being easily trapped into a local optimum and presents a new algorithm which proves to be more simply conducted and with wider overall searching capability.

2 Standard PSO

PSO was presented under the inspiration of bird flock immigration during the course of finding food. In PSO, every optimization is taken as a bird in the searching space

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 723–730, 2008.
© Springer-Verlag Berlin Heidelberg 2008

and it is called a particle. Every particle has a fitness value which is determined by target functions and it has also a velocity which determines its destination and distance. All particles search in the solution space for their best positions and the positions of the best particles in the flock. PSO is initially a group of random particles (random solutions), and then optimum solutions are found by repeated searching. In the course of every repetition, a particle will follow two bests to renew itself, the best position a particle positively find is called pbest, and the best position found for the whole flock us called gbest. All particles will determine following steps through the best experiences of individuals and their companions

$$V_{id}^{'} = \omega V_{id} + \eta_1 rand()(P_{idb} - X_{id}) + \eta_2 rand()(P_{gdb} - X_{id}) \tag{1}$$

$$X_{id}^{'} = X_{id} + V_{id}^{'} \tag{2}$$

In here: ω is called inertia weight, it is a proportion factor that is concerned with former velocity, $0 < \omega < 1$, and are constants and are called accelerating factors, normally $\eta_1 = \eta_2 = 2$; $rand()$ are random numbers, X_{id} represents the position of particle id ; represents the velocity of particle id ; P_{id} , P_{gd} represent separatively the best position particle has found and the position of the best particles id in the whole flock.

In formula(1), the first part represents the former velocity of the particle, it enables the particle to possess expanding tendency in the searching space and thus makes the algorithm be more capable in overall searching; the second part is called cognition part, it represents the process of absorbing individual experience knowledge on the part of the particle; the third part is called social part, it represents the process of learning from the experiences of other particles on the part of certain particle, and it also shows the mutuality and social cooperation among particles.

The flow of PSO is briefly described as: First, to initialize a group of particles, e.g. to give randomly every particle an initial position Xi and an initial velocity Vi, and then to calculate its adaptability value f. In every repetition, to evaluate a particle's adaptability value by analyzing the velocity and positions of renewed particles in formula (1) and (2). When a particle finds a better position than that it has experienced previously, it will mark this coordinate into vector P1, the vector difference between P1 and the present position of the particle will randomly be added to next velocity vector, so that the following renewed particles will search around this point, it's also called in formula(1) cognition fraction. The difference between added vector of the present position of the particle flock and the best position of the flock P_{gd} will also be added to velocity vector, so that the following velocity is adjusted. This is also called in formula (2) social fraction. These two adjustments will enable particles to search around two bests.

The most obvious advantage of PSO is that the convergence speed of the flock is very high, scholars like Clerc [6] has presented facts on the convergence. Here a fatal weakness may result from this characteristic. With constant increase of repetition, the velocity of particles will gradually diminish and reach zero in the end. At this time, the whole flock will be converged at one point in the solution space. If gbest particles haven't found gbest, the whole flock will be trapped into a local optimum; still, the flock's capacity of jumping out of a local optimum is rather weak. The probability of

the occurrence is especially high so far as multihump functions are concerned. In order to get through the disadvantage, this paper presents a new PSO algorithm and applies it to the weight optimization.

3 A New PSO Algorithm

In PSO, the convergence speed of particles is fast, but the adjustments of cognition fraction and social fraction make particles search around P_{gd} and P_{id} all the time. According to velocity and position renewal formula, once pbest in the flock is trapped into a local optimum, the information mutuality mechanism in PSO will attract other particles to approach this local optimum gradually, and in the end the whole flock will be converged at this position. But according to velocity and position renewal formula (1), once the whole flock is trapped into a local optimum, its cognition fraction and social fraction will become zero in the end; still, because and with increase of repetition, the velocity of particles will become zero in the end, thus the whole flock is hard to jump out of the local optimum and has no way to achieve overall optimum. In order to avoid being trapped into a local optimum, the new PSO adopts a new information mutuality mechanism. We all know that when a certain particle is searching in the solution space, it doesn't know the exact position of the optimum solution. But we can not only record the best positions an individual particle and the whole flock have experienced, we can also record the worst positions an individual particle and the whole flock had experienced, thus we may make individual particles move in the direction of evading the worst positions an individual particle and the whole flock have experienced, this will surely enlarge the overall searching space of particles and enable them to avoid being trapped into a local optimum too early, in the same time, it will improve the possibility of finding gbest in the searching space. The new strategy velocity and position renewal formula are as follows:

$$V_{id}^{'} = \omega V_{id} + \eta_1 rand()(X_{id} - P_{idw}) + \eta_2 rand()(X_{id} - P_{gdw}) \tag{3}$$

$$X_{id}^{'} = X_{id} + V_{id}^{'} \tag{4}$$

In here: P_{idw}, P_{gdw} represent the worst position particle id has found and the worst positions of the whole flock.

In standard PSO, the following flying direction of every particle is nearly definite; it can fly to the best individual and the best individuals for the whole flock. From the above conclusion we may easily be aware of the danger it being trapped into a local optimum. In order to decrease the possibility of being trapped, the new PSO introduces genetic alternative strategy: To set particle number in the flock as m, father body and son generation add up to $2m$. To select randomly q pairs from m; as to every individual particle i, if the adaptability value of i is smaller than its opponents, i will win out and then add one to its mark, and finally select those particles which have the maximum mark value into the next generation. The experiments conducted show that this strategy greatly reduces the possibility of being trapped into a local optimum when solving certain functions. The flow of the new PSO is as follows:

Step 1: to initialize randomly the velocity and position of particles;
Step 2: to evaluate the adaptability value of every particle;
Step 3: as to every particle, if its adaptability value is smaller than the best adaptability value P_{idb}, renew the best position P_{idb} of particle id; or else if its adaptability value is bigger than the worst adaptability value P_{idw}, renew P_{idw};
Step 4: as to every particle, if its adaptability value is smaller than the best whole flock adaptability value P_{gdb}, renew the best adaptability value P_{gdb} of particle id; or else if bigger than the worst whole flock adaptability value P_{gdw}, renew P_{gdw};
Step 5: as to every particle,

1) to produce new particle t by applying formula (1) (2),
2) to produce new particle t' by applying formula (3) (4),
3) to make a comparison between t and t' and then select the better one into the next generation;

Step 6: to produce next generation particles according to the above genetic alternative strategy;
Step 7: if all the above steps satisfy suspension needs, suspend it; or turn to Step 3.

4 Weight Optimization Model of College Student Evaluation

This is always a key problem in the works on college student that evaluating the students' comprehensive quality. The early investigation indicates that the results have some limitations. On the one hand, it can not evaluate the students' comprehensive qualities and can not fit the education for full development, because it only evaluates moral, intellectual and physical qualities but lacks cultural quality, psychological quality and creative ability. on the other hand, the evaluation system, which does not use new network techniques, is static, so it can only does after the event and can not give advice to students according to every student's situation.

Through investigating and analyzing many instances, considering the characteristics of China University of Geosciences and the development direction of computer science and technology, the evaluation criteria are divided into five sections: Moral quality aspect, include political idea, social morality, collective idea, discipline idea, socially useful labour and behavior criterion. Professional quality aspect, include the achievements of each courses in this academic year. Physical quality aspect, include recreational and sports activities, gym achievement and physical quality. Psychological quality aspect, include adaptive ability, pressure-resistant capacity, coordinate ability. Developmental quality aspect, include cooperative spirit, practical ability, innovation ideal and ability, management ability, skill and specialty. In order to fairly assess students, the need for objective evaluation of the importance of indicators that the indicators in the index system of weights. Level indicators to determine the weights as an example for discussion, the level indicators secondary indicators determination of similar weight. We think the weight of Moral quality is x1, Professional quality is x2, physical quality is x3, Psychological quality is x4, Developmental quality is x5, now the problem is look for a group of non-negative weights x1, x2, x3, x4, x5, meet x1+ x2+ x3+ x4+ x5=1. Then we get the weight optimization model like the formula as following:

$$\begin{aligned} \min f(x_1,...,x_{K-1}) = & t\sum_{p=1}^{P}\sum_{k=1}^{K-1} | x_{p,k} - x_k | + s\sum_{q=1}^{Q}\sum_{k=1}^{K-1} | x_{q,k} - x_k | + \\ & t\sum_{p=1}^{P} | x_{p,k} - 1 + \sum_{k=1}^{K-1} x_k | + s\sum_{q=1}^{Q} | x_{q,k} - 1 + \sum_{k=1}^{K-1} x_k | \\ & st. \quad a_K \le 1 - (x_1 + x_2 + ... + x_{K-1}) \le b_K \end{aligned} \tag{5}$$

In this formula, the decision-making space is: $0 \le a_k \le x_k \le b_k \le 1, k = 1,2,...,K-1$. K is the number of index, P is the number of teacher participating the weight judge, Q is the number of student participating the weight judge, s is teacher's authority coefficient, t is student's authority coefficient, x_p, k is the weight of the k index determined by the p teacher, x_q, k is the weight of the k index determined by the p student, a_k is the maximum weight of k index, b_k is the minimum weight of k index.

5 Experiments and Results

We use the new algorithm solve the weight optimization. The problem's mathematical description is like formula (5), the sample date come from the question survey and individual estimates for the student of department of Computer Science and Technology, China university of Geosciences. To test the efficiency of the new algorithm, we use the new algorithm running in the computer and to get the index's average value and variance in Table 1-Table 5 based on sample data, and compared with PSO.

Table 1. Average value and variance of the first level indexs

	Average (PSO)	Average (new algorithm)	Variance (PSO)	Variance (new algorithm)
Moral quality	0.2485	0.2445	0.0218	0.0118
Professional quality	0.5485	0.5455	0.0159	0.0119
Physical quality	0.0525	0.0505	0.0019	0.0009
Psychological quality	0.0506	0.0596	0.0015	0.0005
Developmental quality	0.0999	0.0999	0.0022	0.0002

Table 2. Average value and variance of the second level indexs(Moral quality)

	Average (PSO)	Average (new algorithm)	Variance (PSO)	Variance (new algorithm)
political idea	0.1035	0.1005	1.85e-005	1.65e-005
social morality	0.2000	0.2000	2.49e-006	2.47e-006
collective idea	0.1998	0.1998	3.99e-006	3.97e-006
discipline idea	0.1998	0.1998	6.73e-006	6.63e-006
socially useful labour	0.2040	0.2000	2.38e-006	2.36e-006
behaviors	0.1000	0.1000	1.49e-005	1.43e-005

Table 3. Average value and variance of the second level indexs(Physical quality)

	Average (PSO)	Average (new algorithm)	Variance (PSO)	Variance (new algorithm)
recreational and sports activities	0.3038	0.3008	0.0042	0.0012
gym achievement	0.3592	0.3492	0.0062	0.0012
physical quality	0.3504	0.3500	0.0008	0.0000

Table 4. Average value and variance of the second level indexs(Psychological quality)

	Average (PSO)	Average (new algorithm)	Variance (PSO)	Variance (new algorithm)
adaptive ability	0.2508	0.2500	2.7e-005	2.3e-005
pressure-resistant capacity	0.3510	0.3500	2.4e-006	2.2e-006
coordinate ability	0.4004	0.4000	2.5e-005	2.1e-005

Table 5. Average value and variance of the second level indexs(Developmental quality)

	Average (PSO)	Average (new algorithm)	Variance (PSO)	Variance (new algorithm)
cooperative spirit	0.1516	0.1512	0.0015	0.0014
practical ability	0.1504	0.1500	0.0000	0.0000
innovation ideal and ability	0.1999	0.1999	0.0001	0.0001
management ability	0.1501	0.1501	0.0001	0.0001
skill	0.1999	0.1998	0.0001	0.0001
specialty	0.1499	0.1489	0.0013	0.0012

The index weight calculated by the new algorithm can reflect the comprehensive quality of the students in the department of Computer Science and Technology well when carry out the test of students' comprehensive quality. It can also design corresponding evaluation index according to concrete conditions of different departments, then implement evolutionary computation based on the given test value, of course, it still reflect the comprehensive quality of the students well. Compare to the given test value, the index weight calculated obtained by the new algorithm can reflect the justice better when calculate the students' comprehensive achievement.

5 Conclusion

This paper has done two improvements: 1. By introducing a new information mutuality mechanism, we intend to make particles be void of movine in the direction of the worst individual positions and the worst whole flock positions, thus enlarge overall searching space and reduce the possibility of particles to be trapped into a local optimum; 2. By introducing genetic alternative strategy, we have decreased the possibility of being trapped into a local optimum.

Apart from the preservation of the privileges of PSO, the new algorithm enlarges the searching space and the complication level of it is not high. By analyzing the testing results of weight optimization, we reach the conclusion that the new algorithm is efficiency for the function optimization. The new algorithm is more efficient than PSO in coping with optimization problems.

References

[1] Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
[2] Clere, M., Kennedy, J.: The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Trans. on Evolutionary Computation 6(1), 58–73 (2002)
[3] Coello, C.A., Lechuga, M.S.: Mopso: A proposal for multiple objective particle swarm optimization. In: IEEE Proceedings World Congress on Computational Intelligence (CEC 2000), pp. 1051–1056 (2002)
[4] Kennedy, J.: The particle swarm: social adaptation of knowledge. In: Proc. IEEE int. Conf. on evolutionary computation, pp. 3003–3008 (1997)
[5] Ozcan, E., Mohan, C.K.: Analysis of A Simple Particle Swarm Optimization System. Intelligence Engineering Systems Through Artificial Neural Networks, 253–258 (1998)
[6] Clerc, M., Kennedy, J.: The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space. IEEE Trans. on Evolutionary Computation 6, 58–73 (2002)
[7] van den Bergh, F.: An Analysis of Particle Swarm Optimizers. PhD thesis, Department of Computer Science, University of Pretoria, South Africa (2002)
[8] Biswas, Ranjit: An application of fuzzy sets in students evaluation. Fuzzy sets and Systems 74(2), 187–194 (1995)
[9] Robert, S.: The under determination of instructor performance by data from the student evaluation of teaching. Economics of Education Review 21(3), 287–294 (2002)
[10] Chen, S.-M., Lee, C.-H.: New methods for students' evaluation using fuzzy sets. Fuzzy Sets and Systems 104(2), 209–218 (1999)

Combination of Global and Local Search for Real Function Optimization*

Xinsheng Lai

Department of mathematics and computer, Shangrao Normal university, Shangrao, China
xsl2001_jx@163.com

Abstract. This article introduces a global optimizer based on combination of global and local search (OBCGL). OBCGL contains two populations, one for global search and the other for local search. OBCGL utilizes hill climbing method to select new individuals to form new generation instead of selection methods used in Genetic Algorithms (GAs). The algorithm's performance was studied using a test bed of real-valued functions with different degree of multi-modality. In all cases studied, we found that with two populations OBCGL works well.

Keywords: Genetic algorithms, differential evolution, global optimization.

1 Introduction

Genetic Algorithms [1], Evolutionary Strategies [2], and Simulated Annealing [3, 4] are popular and important paradigms in the solution of optimization problems and development of new algorithms. As the complexity of applications increases, there is a corresponding need for more sophisticated global optimization algorithms. One direction is the creation of new global optimizer [5, 6, 7]. Storn and Price created a new real-valued function optimization algorithm called differential evolution, which is fairly fast and reasonably robust. Irizarry came up with a new global optimization algorithm called LARES, which is based on an artificial chemical process. Karaboga and Basturk proposed an optimization based on the intelligent behavior of honey bee swarm. Another direction is the modification of existing optimization algorithms [8, 9]. Zhang and Szeto introduced a modified genetic algorithm, whose mutation probabilities are automatically determined by a function of time, fitness ranking of chromosome, and locus. Nga and Szeto extended the idea behind adaptive mutation probability to include the crossover process. OBCGL belongs to the modifications of GAs.

Genetic Algorithms, one of the most favorite artificial intelligent optimization techniques, uses stochastic transition rules to guide the search [1]. Although genetic algorithm has proved to be versatile and effective optimization algorithm, some problems are still waiting for solving. Genetic algorithms lack hill-climbing capacity; it

* Supported by Nature Science Foundation of Jiangxi Education Committee, P.R. China under Grant Number: GJJ08466 and Jiangxi Province Department of Education, P.R.China under Grant Number: GJJ08467.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 731–739, 2008.
© Springer-Verlag Berlin Heidelberg 2008

easily falls in a trap and finds a local minimum not the true solution. And convergence speed needs improving.

Many works that improve the search performance of GAs have been proposed. Some of them added in other heuristic search algorithm as sub-process of genetic algorithm called Hybrid GA. The most common forms of hybrid GA are to incorporate one or more of hill climbing and/or neighborhood search [10, 11], optimization methods [12, 13], local search [14] or elitist strategy [15, 16, 17] as an add-on extra to the simple GAs loop of recombination and selection. The purpose of hybridization is based on the balance of the global exploration and the local exploitation.

This work aims at introducing a new heuristic Optimizer. The new optimizer searches solution space alternately with global search process and local search process. The hill-climbing heuristic algorithm was used to single out the better solutions to form the next generation. The local search process search within the space near the best solution found so far.

Studied on optimization of several classical multi-modal real valued functions, it turned out that OBCGL outperformed GAs and beared comparison with DE in some studied cases.

2 Description of OBCGL

OBCGL contains two populations of solutions vectors to perform global and local search respectively. A solution vector $X=(X_1, X_2, \ldots, X_d)^T$, consisting of d real numbers, is a pointer of solution space, where d is the dimensionality of solution space. We call the one population used for global search gPopulation and the other used for local search, lPopulation. The local population, namely, lPopulation's local search is led by the best solution found so far by OBCGL, which is recorded into a vector named bSolution. And the global population, namely, gPopulation searches globally in the whole solution space. Figure 1 shows the outline of OBCGL.

Outline of OBCGL

1. Initialize bSolution, gPopulation and lPopulation;
2. If bG is better than bSolution
   then bSolution= bG;
3. If bL is better than bSolution
   then bSolution=bL;
4. While not terminate do
   Local search;
   Global search;
   End while

Fig. 1. The outline of OBCGL. The bSolution is the best solution vector found so far. The bG and bL are the best solution vectors of global population and local population respectively.

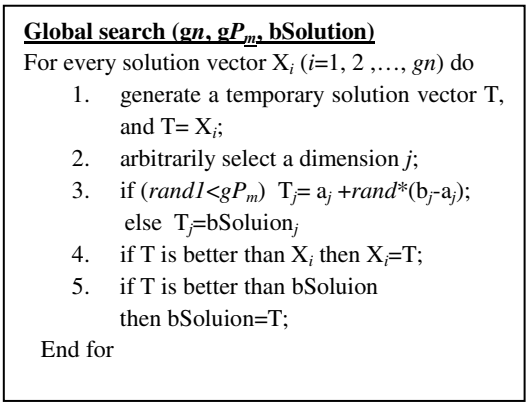


Fig. 2. Global search algorithm. The parameter *gn* is the global population size. The gP_m is the probability of mutation for global population. And the bSolution is the best solution vector found so far. The real numbers *rand1* and *rand* are uniform distributed random numbers generated between 0 and 1. The a_j and b_j are the lower and upper boundaries of the *j*th dimension of the solution space respectively.

Figure 2 shows the process of the global search. The global search performs search within the whole solution space. Global search also inherits some gene loci from bSolution with probability of 1-gP_m. At the same time, the new solution is adopted into gPopulation to replace the old solution considered currently only if the new one is better than the old one.

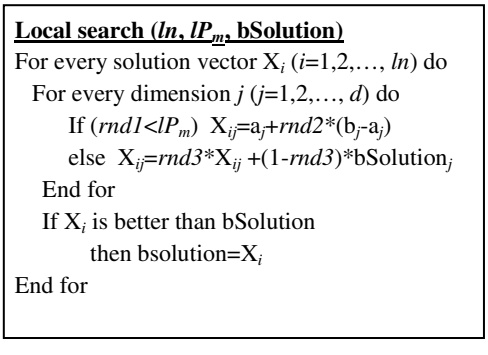


Fig. 3. Local search algorithm. The parameter *ln* is the local population size. The lP_m is the probability of mutation for local population. And the bSolution is the best solution vector found so far. The *d* is the dimensionality of solution space. The real numbers *rnd1*, *rnd2* and *rnd3* are uniform distributed random numbers generated between 0 and 1. The a_j and b_j are the lower and upper boundaries of the *j*th dimension of the solution space respectively.

Figure 3 shows the algorithm of local search. The local search is led by the best solution bSolution found so far. All the solution vectors of local population is move to

the bSolution stochastically. Considering a solution vector X_i, it moves stochastically to bSolution, the new solution vector T generated by this process is:

$$T_j = X_{ij} + rand*(\ bSolution_j - X_{ij}), \quad j=1, 2, \cdots, d. \tag{1}$$

Where *rand* is a uniform random number between 0 and 1.

3 Parameter Settings and Test Bed of Real-Valued Functions

In order to test the performance of OBCGL, we compared OBCGL with GA and DE. In all simulations, the parameters of OBCGL were set to be the same as following: $gn=ln=2$, $gP_m=0.9$, $lP_m=0.0001$.

The GA's parameters were: the population size 10, crossover probability 0.8, and mutation probability 0.005.

The DE's parameters used in this article were: the population size NP=60, selecting weight factor F=0.5, and crossover constant CR=0.8.

Table 1 shows the test bed of real-valued functions.

Table 1. Test bed of real-valued functions

Name	Function	Ranges
Rastrigin	$f_{\text{Rastrigin}}(x) = 10\,n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi\, x_i)\right)$	$[-5.12,5.12]^n$
Schwefel	$f_{\text{Schwefel}}(x) = 418.9828872724\,339 \cdot n - \sum_{i=1}^{n} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	$[-500,500]^n$
Griewangk	$f_{\text{Griewangk}}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600,600]^n$
Ackley	$f_{\text{Ackley}}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi\, x_i)\right) + 20 + e$	$[-30,30]^n$

4 Simulative Experiments and Results

In order to assess the usefulness and stability of OBCGL, the OBCGL algorithm's performance was studied under a test bed of real-valued functions and compared with Genetic Algorithm and Differential Evolution. The "Value" axis stands for the average best-so-far function value, and the "FE" axis stands for the number of function evaluations.

Figure 4 shows the comparison of performances of GA, DE and OBCGL on 30 dimensional Ackley's function within 500 000 FEs. Table 2 shows the best solutions found by them after 500 000 FEs. Figure 4 indicates that the OBCGL converges faster than DE, yet more slow than GA. However, after 500 000 FEs, DE found better solution than OBCGL and GA and converged. After 1 449 601 FEs OBCGL converged and found a better solution than that found by GA after 6 005 730 FEs when converged.

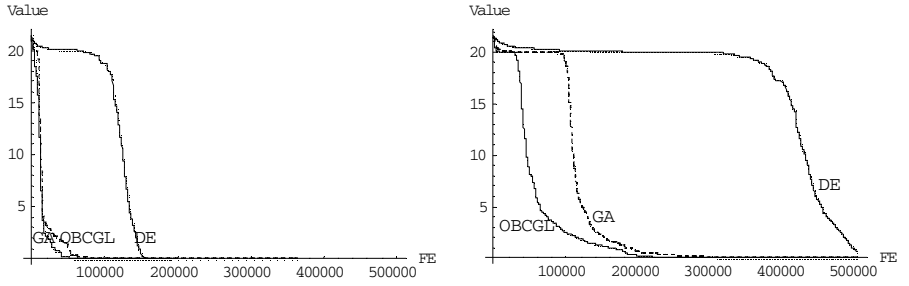


Fig. 4. Average best-so-far function value curves for OBCGL, GA and DE on the multi-modal Ackley's function with 30 (the left one) and 100 (the right one) dimensionalities. 'FE' stands for function evaluations and 'Value' for best function value found so far.

The solution found by OBCGL is better than that found by GA and worse than that found by DE after 500 000 FEs. After converged, GA and OBCGL have found solutions with the same value of 5.225483674e-13, which is close to the value of the solution found by DE at convergence. However, the FEs before convergence of GA is much more than OBCGL, and the FEs for DE to convergence is least.

Figure 4 also shows the convergence curves of OBCGL, GA and DE on the 100 dimensional Ackley function within 500 000 FEs. Table 2 shows the best solutions found by them after 500 000 FEs. The right one of figure 4 illustrates that OBCGL converges fastest among OBCGL, DE and GA whereas DE converges to the globally best solution at a most speed at the early stage. However, after 2 066 400 FEs DE has converged to a solution with value of 5.083375127e-13 near to the globally best solution, and after 18 946 081 FEs OBCGL has also converged to a solution with value of 6.04260782e-13 near to the globally best solution. After 22 338 650 FEs when converged, GA has reached a solution with value of 1.285497319e-6 far from the globally best solution.

The solution found by OBCGL is better than that found by DE and GA after 500 000 FEs. However, after convergence, the solution found by OBCGL is worse yet quite close to that found by DE and much better than that found by GA. Though it took much more FEs to converge for OBCGL, and the FEs took to converge for GA even more.

The left one of figure 5 illustrates that OBCGL converges at medium speed comparing with DE and GA. After 500 000 FEs and on 30 dimensional Griewangk's function, DE has converged to the globally best solution, and OBCGL found a solution with value of 1.012048751e-5, yet GA only found a solution with value of 0.03436209634. Even when it converged at FE=14 049 950, GA only found a solution valued 0.007396041881. OBCGL converged at FE=636 481, and found a solution valued 0.

With respect to 100 dimensional Griewangk's function, after 500 000 FEs DE has converged to the globally best solution, and OBCGL found a solution valued 1.795183813e-8, yet GA found a solution only valued 0.05876326934. Even converged, GA has come to a solution valued 0.0245733168. However, OBCGL and DE have reached the globally best solution valued 0.

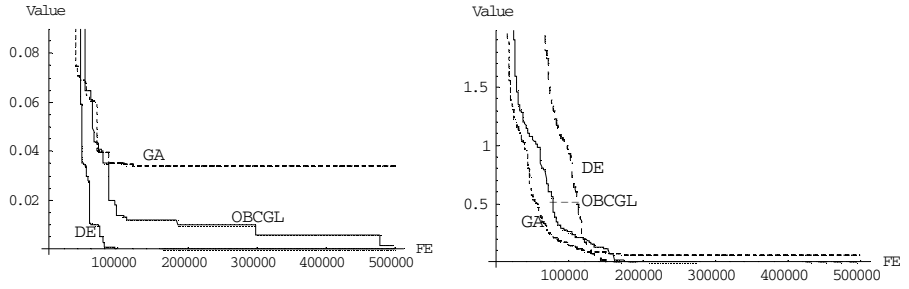


Fig. 5. Average best-so-far function value curves for OBCGL, GA and DE on the multi-modal Griewangk's function with 30 (the left one) and 100 (the right one) dimensionalities

The left one of figure 6 indicates that OBCGL converges at a medium speed comparison to DE and GA on the 30 dimensional Rastrigin's function. DE converges fast and GA converges most slowly. On 30 dimensional Rastrigin's function, after 500 000 FEs, as shown in table 2, the value of the best solution found by OBCGL is 9.549694369e-12, near to the globally best value 0, whereas the value of the best solution found by GA is only 0.00146484375, far away from 0, and DE has found the globally best solution. GA converged at a solution valued 3.051757813e-5 after 1 501 440 FEs, but OBCGL could converged to a solution valued 5.684341886e-14 within 755 521 FEs.

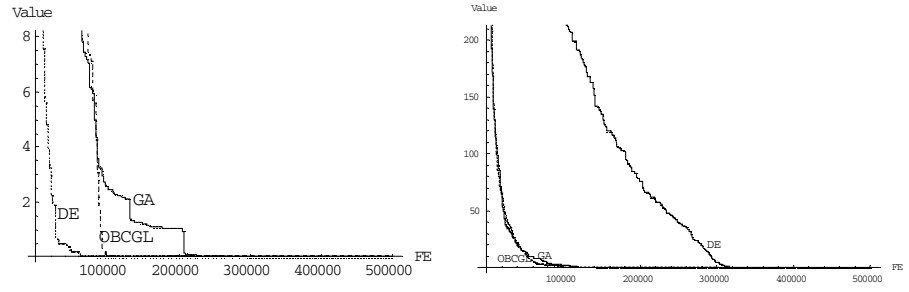


Fig. 6. Average best-so-far function value curves for DHGA, GA and DE on the multi-modal Rastrigin's function with 30 (the left one) and 100 (the right one) dimensionalities. FE stands for function evaluations and Value for best function value found so far.

The right one of figure 6 illustrates that OBCGL converges more rapidly than DE and GA on 100 dimensional Rastrigin's function. As shown in table 2, after 500 000 FEs, the value of the best solution found by OBCGL has come to 8.99300403e-8, however the value of the best solution found by GA is only 0.00048828125, and DE has come to 1.265561878e-9. After converged at FE=7 656 361, OBCGL found a solution with value of 9.094947018e-13, and DE has come to the globally best solution when converged at FE=592 680, whereas, GA converged at FE=8 858 420 and found the solution only valued 0.0000831054688.

On both 30-dimensional and 100-dimensional Rastrigin's function, the solution found by OBCGL is better than that found by GA and worse than DE after both 500 000 FEs and convergence.

The left one of figure 7 shows the convergence curves of OBCGL, GA and DE on 30-dimensional Schwefel's function. It illustrates that GA and DE converge faster than OBCGL. But as shown in table 2 the solutions found by them after 500 000 FEs are quite different. After 500 000 FEs, the value of the best solution found by DE is 3.637978807e-12, and the value of the best solution found by OBCGL is 8.321177847e-5, however the best solution found by GA is only valued 0.005172860343. OBCGL has come to a solution valued 7.275957614e-12 when converged at FE=2 712 841, whereas GA found a solution with value of 0.000114789841 when converged at FE=2 011 410.

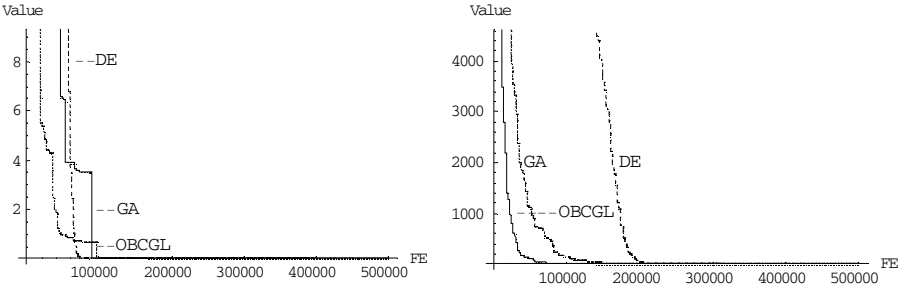


Fig. 7. Average best-so-far function value curves for OBCGL, GA and DE on the multi-modal Schwefel's function with 30 (the left one) and 100 (the right one) dimensionalities. FE stands for function evaluations and Value for best function value found so far.

Table 2. Best solutions found by GA, DE and OBCGL after 500,000 FEs on the Schwefel's function within solution space of $[-500, 500]^{30}$ and $[-500, 500]^{100}$. 0 means the value of solution is below 1e-25.

function	GA	DE	OBCGL	GA	DE	OBCGL
	30 dimensionalities			100 dimensionalities		
Ackley	5.338345659e-7	4.94126658e-13	7.832749836e-8	0.01562644355	0.6168918167	0.0005257132085
Griewangk	0.03436209634	0	1.012048751e-5	0.05876326934	0	1.795183813e-8
Rastrigin	0.00146484375	0	9.549694369e-12	0.00048828125	1.265561878e-9	8.99300403e-8
Schwefel	0.005172860343	3.637978807e-12	8.321177847e-5	0.5543522239	1.236912794e-10	0.02469178396

The right picture of figure 7 shows the convergence curves of OBCGL, GA and DE on 100-dimensional Schwefel's function. It illustrates that OBCGL converges faster than DE and GA. After 500 000 FEs, as shown in table 2, the value of the best solution found by DE is 1.236912794e-10, and the value of the best solution found by OBCGL is 0.02469178396, however the best solution found by GA is only valued 0.5543522239. OBCGL has come to a solution valued 1.455191523e-10 when

converged at FE=2 636 161, whereas GA found a solution with value of 0.01528974343 when converged at FE=5 829 010.

On both the 30 dimensional and the 100 dimensional Schwefel's function, the solution found by OBCGL is better than that found by GA and worse than DE after both 500 000 FEs and convergence.

5 Conclusions and Further Research

In this work, a new general-purpose real function optimization algorithm was introduced, whose performance in terms of robustness and efficiency was studied with a set of challenging benchmark problems. The OBCGL algorithm performed very well, converging to near global optimal solutions when solving different class of problems with different degree of multi-modality. In all cases studied, OBCGL was on average faster than GA and more robust in finding the global optimal solution. In some cases studied, in terms of solution quality and convergence speed, performance of OBCGL was close to and even the same as that of DE, which was test to be good optimizer for real function optimization.

Further research will include theoretical analysis to better understand this algorithm's convergence properties. Above of all, further efforts will still be taken to further improve the performance of OBCGL.

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
2. Schwefel, H.P.: Evolution and Optimum seeking. John Wiley, Chichester (1995)
3. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated Annealing. Science 220, 671–680 (1983)
4. Ingber, L.: Simulated annealing: Practice versus theory. J. Mathl. Comput. Modelling 18, 29–57 (1993)
5. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
6. Irizarry, R.: LARES: An Artificial Chemical Process Approach for Optimization. Evolutionary Computation Journal 12, 435–460 (2004)
7. Karaboga, D., Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization 39, 459–471 (2007)
8. Zhang, J., Szeto, K.Y.: Mutation matrix in evolutionary computation: Anapplication to resource allocation problem. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 112–119. Springer, Heidelberg (2005)
9. Law, N.L., Szeto, K.Y.: Adaptive Genetic Algorithm with Mutation and Crossover Matrices. In: IJCAI 2007, pp. 2330–2333 (2007)
10. Yamada, T., Reeves, C.R.: Solving the Csum permutation flowshop scheduling problem by genetic local search. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 230–234 (1998)
11. Hwang, S.F., He, R.S.: Improving real-parameter genetic algorithm with simulated annealing for engineering problems. Adv. Eng. Softw. 37, 406–418 (2006)

12. Wang, L.: A hybrid genetic algorithm-neural network strategy for simulation optimisation. Applied Mathematics and Computation 170, 1329–1343 (2005)
13. Kido, T., Kitano, H., Nakanishi, M.: A Hybrid Search for Genetic Algorithms: Combining Genetic Algorithms, Tabu Search, and Simulated Annealing. In: Proceedings of the 5th International Conference on Genetic Algorithms, San Mateo, CA, pp. 641–647 (1993)
14. Freisleben, B., Merz, P.: A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Travelling Salesman Problems. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 159–164 (1996)
15. Murata, T., Ishibuchi, H., Tanaka, H.: Genetic Algorithms for Flowshop Scheduling Problems. Computer & Industrial Engineering 30, 1061–1071 (1996)
16. Warattapop, C., Peeraya, T., Pupong, P.: A New Heuristic for Improving the Performance. Proceedings of world academy of science, engineering and technology 21, 217–220 (2007)
17. Liu, J., Zhong, W.C., Liu, F., Jiao, L.C.: An Organizational Evolutionary Algorithm for constrained and Unconstrained Optimization Problems. Chinese Journal of computers 27, 157–167 (2004)

Multi-Classifier Systems (MCSs) of Remote Sensing Imagery Classification Based on Texture Analysis

Hongfen Li, Guangdao Hu, and Jiang-feng Li

Institute of Mathematical and Remote Sensing Geology,
China University of Geosciences,
Wuhan 430074, China
lihongfen_dida@163.com

Abstract. This article concerns methods of improving the accuracy of land cover maps using Very High-resolution Satellites (VHRS).It discusses two methods for increasing the accuracy of classifiers used in land cover mapping. One is texture analysis using GLCM method and the other is multiple classifier system (MCSs) using voting rules. A case study of QuickBird Imagery of an area in Chenggong County of Yunnan Province is conducted based on an analysis of QuickBird imagery. The experiment results show that these two methods can improve the accuracy greatly. The applying of texture bands makes an increase of 2.6816%, and the MCSs make an increase of 3.9512%.

Keywords: multiple classifier system (MCSs), co-occurrence probability, texture analysis, land-cover classification.

1 Introduction

Currently, Very high-resolution satellites (VHRS) including QuickBird provide a useful way for us to periodically monitor detailed land use in broad areas. Pixel sizes on the ground of VHRS sensors are small enough to capture geometrical details of common land use patches, and most geographic textures on VHRS images.

Compared with medium/low resolution image, VHRS imagery gets more texture characteristics and geographic structure information. But the conventional classification algorithms based on pixel spectrum often get low accuracy.

The conventional computer classification methods focus on the following 6 aspects: 1) the classification based on mathematic statistics; 2) the classification based on imagery features; 3) the classification of RS imagery of single source of sensor; 4)the classification based on single pixel; 5)hard classification that every pixel must belongs to a particular class; 6)one classifier throughout the classification; the limitation of these technologies can not make a more accurate result, new method mainly do some improvement on these 6 aspects.

In this paper, two kinds of solutions are proposed to improve the classification accuracy of VHRS, 1) texture analysis using GLCM method; 2) Multiple classifier systems (MCSs) with a voting rule. And it is arranged in the following manner, section 2 details the algorithms used to capture co-occurrence probability texture features; section 3 analysis the experimental results, Section 4 give an overall conclusion of the whole paper.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 740–749, 2008.
© Springer-Verlag Berlin Heidelberg 2008

2 Method

2.1 GLCM Texture Feature Extraction

Texture reflects the surface roughness and is little influenced by the outside. In this paper, 8 texture feature of Gray level co-occurrence matrix (GLCM), and summarized its characteristic from window size and the texture reflectance. Currently, the texture analysis methods in common use can be categorized into four groups: statistical, geometrical, model-based and signal processing-based.

Grey-level co-occurrence matrices (GLCMs), also called grey level co-occurrence probability (GLCP) method, developed by Haralick er al. (1973), is a popular method for determining texture within remotely sensed digital imagery.

The GLCM technique employs the following steps. The probability of co-occurrence between two grey levels i and j given a relative orientation (y) and distance (d) can be computed for all possible co-occurring grey level pairs in an image window. The GLCM stores these probabilities and, as such, is dimensioned to the number of grey levels available. Then, selected statistics are applied to the GLCM by stepping through the entire matrix (i.e. over all probabilities) to calculate the texture features.

Given a spatial window within the image, the co-occurrence method finds the conditional joint probabilities, C_{ij} , of all pair wise combinations of grey levels given the inter-pixel displacement vector $(\delta x , \delta y)$, which represents the separation of the pixel pairs in the x- and y-directions respectively. The set of grey level co-occurring probabilities (GLCP) can be defined as equation 1.

$$C_{ij} = P_{ij} \Big/ \sum_{i,j=0}^{G-1} P_{ij} \tag{1}$$

$$\text{Mean: } f_{MEA=} \sum_{i=0}^{M} \sum_{j=0}^{N} i \times p(i,j) \tag{2}$$

$$\text{Contrast: } f_{CON=} \sum_{i=0}^{M} \sum_{j=0}^{N} (i-j)^2 p(i,j) \tag{3}$$

$$\text{Entropy: } f_{ENT=} -\sum_{i=0}^{M} \sum_{j=0}^{N} p(i,j) \log p(i,j) \tag{4}$$

$$\text{Inverse difference moment (homogeneity): } f_{INV=} \sum_{i=0}^{M} \sum_{j=0}^{N} \frac{p(i,j)}{1+(i-j)^2} \tag{5}$$

$$\text{Correlation: } f_{COR=} \sum_{i=0}^{M} \sum_{j=0}^{N} \frac{(i-u)(j-u)p(i,j)}{\sigma^2} \tag{6}$$

$$\text{Angular second moment: } f_{ASM} = \sum_{i=0}^{M}\sum_{j=0}^{N} p(i,j)^2 \tag{7}$$

$$\text{Variance: } f_{VAR} = \sum_{i=0}^{M}\sum_{j=0}^{N} p(i,j)\times(i-u)^2 \tag{8}$$

$$\text{Dissimilarity: } f_{DIS} = \sum_{i=0}^{M}\sum_{j=0}^{N} |i-j|\, p(i,j) \tag{9}$$

$$u = \sum_{i=0}^{M}\sum_{j=0}^{N} i\times p(i,j)\,, \tag{10}$$

$$\sigma^2 = \sum_{i=0}^{M}\sum_{j=0}^{N} (i-u)^2\times p(i,j) \tag{11}$$

This paper uses co-occurrence measures to apply texture filters that are based on the co-occurrence matrix. These filters include mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment, and correlation. The calculations are showed with equation 2～11. This is a matrix of relative frequencies with which pixel values occur in two neighboring processing windows separated by a specified distance and direction. It is time consuming to calculate about 255 kinds of texture feature combinations based on different parameters such as orientations, distances, and statistics. So firstly we have to find the optimal texture feature combination.

2.2 Multiple Classifier Systems (MCSs)

Multiple classifier systems (MCSs), based on the combination of outputs of a set of different classifiers have been proposed in the field of pattern recognition as a method for the development of high performance classification systems. In this paper, the combination of several classification algorithms is used as a tool for increasing the accuracy and the reliability of the classification maps obtained by each single classifier; classifiers are combined by means of voting rules.

With the complexity of pattern recognition increased and novel algorithms developed, researchers find that although different classifiers have different classification performance, their misclassification set are not consistent with each other. That is, some sample misclassified by one classifier may be recognized by another classifier. Different classifiers are complementary to each other. In this way, the combination of classifiers can improve the accuracy of the classification.

The voting rule, a mathematically and computationally simple method of combining classifiers, was found to perform almost as well as more complex unconstrained stacked regression method and better than the sum and constrained stacked regression methods. The area of combining classifiers is rapidly developing within the fields of statistics and computer science. With the complexity of pattern recognition increased and novel algorithms developed, researchers find that although different classifiers

have different classification performance, their misclassification set are not consistent with each other. That is, some sample misclassified by one classifier may be recognized by another classifier. Different classifiers are complementary to each other.

The result is determined by choosing the class for each pixel by the majority vote of all supervised methods you chose to run. For example, if three classifiers determined a pixel was asphalt and two determined it was concrete, the final result is asphalt. In the case of a tie, if there still are pixels in question, the majority class of the neighboring pixels is used. Define a kernel size and the center pixel in the kernel will be replaced with the class value that the majority of the pixels in the kernel has.

The voting rule is expressed in equation 12～13, where i=1,2,…,M, 0<α<1, K means the threshold value to determine the voting result. As the values of $\alpha$and K change, different classification rules are generated. The stricter the rules are, the higher the reliability is.

$$E(X)=\begin{cases} j & T_E(X\in C_i)=\max_{i\in A}\alpha K \\ M+1 & other \end{cases} \tag{12}$$

$$T_E(X\in C_i)=\sum_{i=1}^{K}T_k(X\in C_i) \tag{13}$$

In this paper, 4 kinds of classification algorithms, including Maximum likelihood, Minimum distance, Mahalanobis distance and Support vector machine, were chosen to compose a multiple classifier systems (MCSs).

3 Experiments

3.1 Researching Area and Image Pre-processing

The data used in this paper were acquired over an agricultural area near the village of Chenggong district in Yunnan province, and was collected in 2003(QuickBird). The image area is about 1757*1895 pixels; located in E102°49′25″~102°50′09″、N24°54′30″~24°55′12″, Fig.1 shows the location of the researching area. The data contains red, green, blue and NIR bands, with an additional panchromatic band. After fusion, the resolution of the image was improved to 0.61m. Before classification; some preprocessing work was done. Geometric correction is performed using 25 GCPs; PCA (principle component analysis) was applied for spectral enhancement; orthorectification of the image is done with 40 GCPs using a Polynomial model, whose RMS is 1.93m. Atmospheric correction was made to the image using Dark-Object-Methods.

Considering the land use types distributed in this area, the classification tests all generate 6 classes, including bare land, road, building, water, forest and arable land.

The same area was processed by every classification method, in order to assess the ability of classifying of the methods. The same training set was used, also a test set for accuracy assessment. Fig.1 shows the location of the researching area.



Fig. 1. The location of study area of Chenggong in Yunnan province

3.2 Texture Analysis

3.2.1 Optimal Window Size Analysis

We calculate the accuracy of 4 original bands and 8 additional texture bands stack together, in order to examine the influence of window size on accuracy of classification. Table 1 shows the accuracy statistics of classifications. From table 1 we can see the accuracy of classification is getting higher while the window size getting larger, but there is no obvious improvement when a window size gets larger than 7×7. Considering the edge effect which becomes more severe when the window size gets larger, in this paper ,we choose a window size of 7×7. The following experiments are performed under the circumstances of 7×7 of window size if without indication.

Table 1. Influence of window size of texture information on classification accuracy

Window size	Total accuracy (%)	kappa
3×3	71.9523	0.6569
5×5	76.1367	0.7060
7×7	80.6409	0.7510
9×9	78.5125	0.7347
11×11	78.8928	0.7395
13×13	69.3363	0.6251
15×15	80.0162	0.7536
17×17	80.0828	0.7546
19×19	70.0332	0.6240

3.3 Optimal Texture Feature Combination

A primary computational drawback of GLCMs is that they require demanding computation time to perform every possible texture feature combination. The key point is to select texture features which are independent and discriminating, and will aid mostly

in the classification process. We calculate the accuracy of 7 different types of texture feature combinations. Researches show that the orientations and distances have weak impact on the classification accuracy[7]. So in this paper we do not discuss conditions of different orientation and distance. Based on the work of David A. Clausi[1] , the most helpful texture feature , will be mean and entropy, while using GLCMs for VHRS imageries. So in this paper, we choose $f_{MEA} + f_{ENT}$ to be the first combination, then make additions and calculate the accuracies. Table2 shows the results. Considering the balance of accuracy and time consuming calculation problem, in this paper, the 5_{th} combination is chosen to be the optimal texture feature combination.

Table 2. Variance accuracy in different texture feature components analysis

Texture feature component	Total accuracy (%)	kappa
$f_{MEA} + f_{ENT}$	80.8610	0.7652
$f_{MEA} + f_{ENT} + f_{VAR}$	80.9307	0.7663
$f_{MEA} + f_{ENT} + f_{VAR} + f_{ASM}$	81.3712	0.7716
$f_{MEA} + f_{ENT} + f_{VAR} + f_{ASM} + f_{CON}$	81.8961	0.7781
$f_{MEA} + f_{ENT} + f_{VAR} + f_{ASM} + f_{CON} + f_{INV}$	84.3268	0.8077
$f_{MEA} + f_{ENT} + f_{VAR} + f_{ASM} + f_{CON} + f_{INV} + f_{DIS}$	84.3258	0.8113
$f_{MEA} + f_{ENT} + f_{VAR} + f_{ASM} + f_{CON} + f_{INV} + f_{DIS} + f_{COR}$	84.6664	0.8118

3.4 Multi-Classifier Systems (MCSs)

To assess the performance of these classifiers, accuracy was estimated using the same training samples collected for land cover mapping. The results in table 3 show that MCSs produced substantial increases in estimated accuracy when compared to every single uncombined classifier. MCSs get a highest accuracy of 91.9992%, the kappa is 0.8999, while the second highest is 88.0480% from SVM classifier. The following research makes a further comparison of classification accuracy between MCSs and SVM for every particular class.

Table 3. Variance of accuracy in different classifier analysis

Classifier	Overall accuracy (%)	kappa
Mahalanobis distance	74.2042	0.6784
Minimum distance	79.4532	0.7407
Maximum likelihood	83.7763	0.7973
Support vector machine(SVM)	88.0480	0.8507
MCSs	91.9992	0.8999

Fig2 (a)-(e) show the results of 5 different classifiers: Mahalanobis distance, Minimum distance, Maximum likelihood, Support vector machine and Voting rule multiple classifier. Fig2 (f) show the result of classification of original 4 bands (without texture bands) using SVM classifier. Table 4～5 show the confusion matrixes and accuracy assessment of MCSs including the texture bands. Table 6 is a comparison result of original image classification (only contain 4 bands: R, G, B, N) with MCSs, and shows a difference in accuracy with or without the texture features.

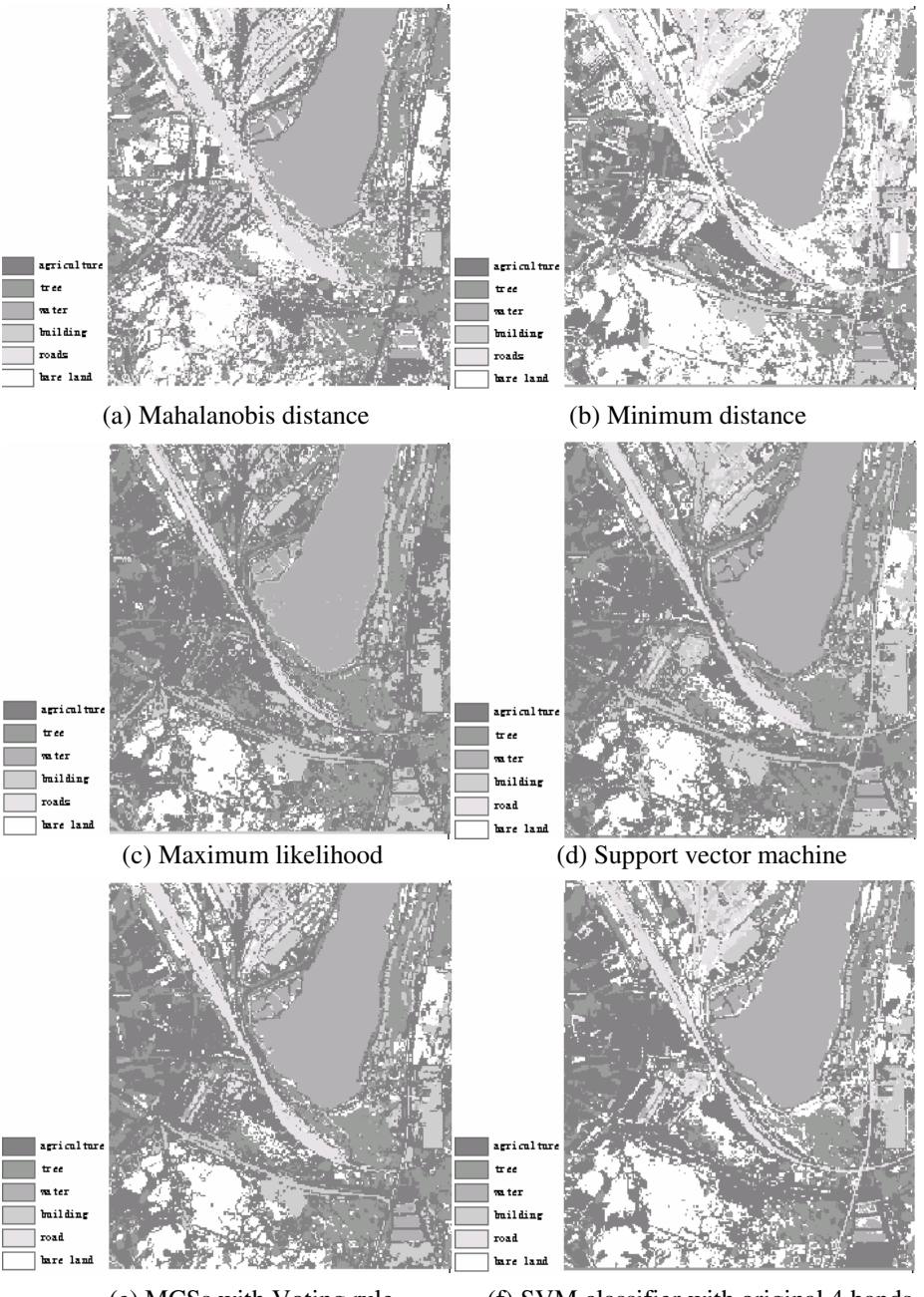


Fig. 2. The classification results of different classifiers and the result without texture bands

Table 4. MCSs classification confusion matrixes

class	agriculture	tree	water	building	road	Bare land	Total
agriculture	53394	636	20	1213	929	3569	59761
tree	2227	36537	471	6	0	911	40152
water	0	11	28023	0	0	0	28034
building	46	2	14	21246	260	0	21568
road	169	7	18	218	19980	55	20447
bare land	8632	1987	55	18	43	56493	67228
total	64468	39180	28601	22701	21212	61028	237190

Table 5. MCSs classification accuracy

class	Commission (%)	Omission (%)	Prod. Acc (%)	Omission (%)
agriculture	10.65	17.18	82.82	89.35
tree	9.00	6.75	93.25	91.00
water	0.04	2.02	97.98	99.96
building	1.49	6.41	93.59	98.51
road	2.28	5.81	94.19	97.72
bare land	15.97	7.43	92.57	84.03

Table 6. The accuracy comparison of several combinations of texture band and classifiers

(Optimal 3*3 window size)	4 original bands(svm)	4 original bands(MCSs)	adding optimal texuture bands(svm)	adding optimal texuture bands(MCSs)
Overall accuracy (%)	84.1363	87.0446	88.0480	91.9992
kappa	0.7997	0.8373	0.8507	0.8999
Agriculture (%)	80.44	75.73	80.88	87.72
Tree (%)	86.11	92.18	89.34	95.18
Water (%)	82.12	93.77	98.08	98.00
Building (%)	90.76	88.96	96.90	95.51
Road (%)	62.81	72.99	88.20	86.46
Bare land (%)	92.66	96.72	86.75	92.29

From the results, we can see that the inclusion of texture bands improves the accuracy remarkably. The accuracy of original 4 bands gets an accuracy of 84.1363%, but texture band improves the result to 88.048%.multiple classifier makes a further improvement to 91.9992%. The applying of texture bands makes an increase of 2.6816% and the MCSs makes an increase of 3.9512%. It shows that the adding of texture band and MCSs make remarkable improvement to classification accuracy. Also, we can find that although the MCSs result (with texture bands) gets the highest overall accuracy, it does not get the highest for every particular class. For example, water, buildings, road all get a decline in accuracy while using MCSs. Meanwhile, the including of texture bands improves classification accuracy for every class also the overall accuracy.

4 Conclusions

1) In this paper, an approach to the automatic design of MCSs formed by different classification algorithms based on voting rule has been described.
2) The experiment results show that GLCMs method can improve the accuracy greatly. However, a primary computational drawback of GLCMs is that they require demanding computational requirements when applying the statistics. Before applying to the data set, we make a series of test to find the optimal window size and the optimal texture feature combination, which is 7×7, so is the texture feature, the 5_{th} combination is chosen.
3) The comparison results shows (in table 6) that the applying of these two methods improves the accuracy of classification greatly, with an increase of 7.8629%.The applying of texture bands makes an increase of 2.6816%, and the MCSs makes an increase of 3.9512%.But not every particular class benefit from MCSs, which mean linear combination rules can only improve the overall accuracy, when a classifier is good at classifying a particular class, but has a low overall accuracy, the specialty cannot be fully used to aid increase the classification accuracy.
4) Classifiers make identical errors on difficult patterns, therefore, the fundamental need for methods aimed at designing accurate and diverse classifiers is currently acknowledged. In further researches, new texture feature extraction methods should be tried to improve the accuracy, such as methods based on wavelet decomposition; fractal lacunarity analysis etc; non-linear classifier combination rules should be developed to make a further improvement of classification accuracy for every particular class, such as SVM, decision tree, neural network, etc.

References

1. Clausi, D.A., Zhao, Y.: Rapid extraction of image texture by co-occurrence using a hybrid data structure. Computer & Geosciences, 763–774 (2002)
2. Hay, G.J., Niemann, K.O., McLean, G.F.: An object-specific image-texture analysis of H-Resolution Forest Imagery. Remote Sensing of Environment 55, 108–122 (1996)
3. Giacinto, G., Roli, F.: An approach to the automatic design of multiple classifier systems. Pattern Recognition letters 22, 25–33 (2001)
4. Wang, L., Liu, J.: Texture classification using multi-resolution Markov random field models. Pattern Recognition letters 20, 171–182 (1999)
5. Brian, M., Steele: Combining Multiple Classifiers: An Application Using Spatial and Remotely Sensed Information for Land Cover Type Mapping. Remote Sensing of Environment 74, 545–556 (2000)
6. Berberoglu, S., Curran, P.J., Lloyd, C.D., Atkinson, P.M.: Texture classification of Mediterranean land cover. International Journal of Applied Earth Observation and Geoinformation 9, 322–334 (2007)
7. Jing, Y.: The application of land-use classification for high resolution remote sensing image based on space characteristic. A Dissertation Submitted to China university of Geosciences for the Degree of Doctor of Engineering
8. Haralick, R.M., Shanmugam, K., Dinstein, I.: Texture features for image classification. IEEE Transactions on Systems, Man, and Cybernetics 3, 610–625 (1973)

9. Kayitakire, F., Hamel, C., Defourny, P.: Retrieving forest structure variables based on image texture analysis and IKONOS-2 imagery. Remote Sensing of Environment 102, 391–401 (2006)
10. Aguera, F., Aguilar, F.J., Aguilar, M.A.: Using texture analysis to improve per-pixel classification of very high resolution images for mapping plastic greenhouses. ISPRS Journal of Photogrammetry & Remote Sensing (2008) doi:10.1016/j.isprsjprs.2008.03.003
11. Johansen, K., Coops, N.C., Gergel, S.E., Stange, Y.: Application of high spatial resolution satellite imagery for riparian and forest ecosystem classification. Remote Sensing of Environment 110, 29–44 (2007)
12. Wang, L., Soursa, W.P., Gong, P., Biging, G.S.: Comparison of IKONOS and QuickBird images for mapping mangrove species on the Caribbean coast of Panama. Remote Sensing of Environment 91, 432–440 (2004)
13. Lowry, J., Ramsey, R.D., Thomas, K., et al.: Mapping moderate-scale land-cover over large geographic areas within a collaborative framework: A case study of the Southwest Regional Gap Analysis Project(SWReGAP). Remote Sensing of Environment 108, 59–73 (2007)
14. Fukunaga, K., Hayes, R.R.: Effects of sample size in classifier design. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 873–885 (1989)
15. Hughes, G.: On the mean accuracy of statistical pattern recognizers. Information Theory 4, 55–63 (1968)
16. Lee Daniel, D., Sebastian, S.H.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
17. Yunhao, C., Peijun, S., Xiaobing, L., et al.: A combined approach for estimating vegetation cover in urban/suburban environments from remotely sensed data. Computer & Geosciences 32(9), 1299–1309 (2006)
18. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Systems Man and Cybernetics 22(3), 418–435 (1992)
19. Kittler, J., Hatef, M., Duin, R.P.W., et al.: On combining classifiers. Transactions on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
20. Vapnik Vladimir, N.: The Nature of Statistical Learning theory, 2nd edn. Springer, New York (2000)
21. Vapnik, V.N.: An overview of statistical learning theory. IEEE Transactions on Neural Networks 10(5), 988–999 (1999)
22. Hunter, E.L., Power, C.H.: An assessment of two classification methods for mapping Thames Estuary intertribal habitats using CASI data. International Journal of Remote Sensing 23(15), 2989–3008 (2002)
23. Settle, J.J., Drake, N.: Linear mixing and the estimation of ground cover proportions. International Journal of Remote Sensing 14(6), 1159–2277 (1993)
24. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. The Journal of Machine Learning Research 5, 975–1005 (2004)
25. Khotanzad, A., Kashyap, R.: Feature selection for texture recognition based on image synthesis. IEEE Transactions on Systems, Man, and Cybernetics 17, 1087–1095 (1987)
26. Tuceryan, M., Jain, A.K.: Texture Analysis. Handbook of Pattern Recognition and Computer Vision, 235–276 (1993)

Particle Swarm Optimization with Dynamic Dimension Crossover for High Dimensional Problems

Chengyu Hu[1,2], Xuesong Yan[1], and Chuanfeng Li[2]

[1] School of Computer, China University of Geosciences, Wuhan, China 430074
huchengyu@cug.edu.cn, yanxs1999@126.com
[2] Department of Control Science and Engineering, Huazhong University of Science & Technology, Wuhan, China 430074
lichuanfeng@sina.com

Abstract. Previous work presented some modified approaches based particle swarm optimization (PSO) to solve complex optimization problems. Preliminary results demonstrated that PSO with crossover (CPSO) constituted a promising approach to solve some optimization problems. However how to optimize high dimensional problem with crossover became challenging. In this paper, a modified PSO with dimension crossover is proposed. First we analyze the cause of hardly optimizing the high dimensional problem, and then design one dynamic dimension crossover PSO (DDC-PSO) to cope with high dimensional problems. Theoretical analysis is also presented to show why the modified algorithm can be effective. Finally DDC-PSO is tested on five benchmark optimization problems and the results show a superior performance compared to the standard PSO and CPSO.

Keywords: PSO, High Dimensional Problems, Dimension Crossover.

1 Introduction

The process of optimization has presented greater importance in many real life engineering problems. Recently, Particle Swarm Optimization (PSO) proposed by Eberhart and Kennedy [1] has gained a huge popularity due to its algorithmic simplicity and effectiveness. However much of the current research focuses on optimizing functions involving relatively few dimensions (typically around 20 to 30, and generally fewer than 100). For high dimensional problems (e.g.≥100), on the other hand, the PSO performance starts to suffer due to the curse of dimensionality [2].

Recently some modified PSO has been applied to these problems. These approaches are basically divided into two categories: one is to reduce the dimensions of high dimensional problems, and use multiple populations to optimize through collaborative evolution. For example, Potter [3] proposed that divided the search space and made a collaborative Optimization, Tony Huang [4] presented a simple and powerful uPSO, which was used to optimize the high dimensional functions by the collaboration of multiple micro-subgroups (each subgroup 3-5 particles), Lei Kaiyou [5], who also divided the whole population into multiple sub-groups, and each subgroup had different inertia weight function, and then cooperatively to optimize benchmark

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 750–759, 2008.
© Springer-Verlag Berlin Heidelberg 2008

functions. The other is hybrid algorithm which is to mix the local search algorithm or the arithmetic operators with PSO. For example, Zhang Jingwei [6] mixed simulated annealing with PSO for high-dimensional optimization. Gao Hao and Li Li [7, 8] used the quantum PSO to cope with the high dimensional problems.

In this paper we investigate the use of PSO for high dimensional problems. It has been found that combining PSO with dimension crossover operator can in many cases significantly improve the performance of PSO. The analysis and proof are given why the new algorithm work well.

The other sections are organized as follows. Section 2 describes the standard particle swarm algorithm along with some of the difficulties encountered with high dimensional optimization problems. Section 3 describes the modifications made to the standard PSO algorithm. Section 4 outlines the experimental settings used in our tests, followed by a discussion of the test results. Finally, section 5 offers some conclusions.

2 Curse of Dimensionality

The high dimensional problem is hard to optimize because of "curse of dimensionality"which refers to the exponential growth of volume associated with adding extra dimensions to a problem space. For population based optimization algorithms such as PSO, this rapid growth in volume means that as the dimensionality of the problem increases, each of the particles has to potentially search a larger and larger area in the problem space.

In the space exist one d dimension hyper sphere with radius r, its volume is:

$$V(B_d(r)) = V(B_d(1)) * r^d, \quad \text{here } V(B_d(1)) = \pi^{\frac{d}{2}} \Big/ \Gamma(\frac{d}{2}+1) \tag{1}$$

If exist one d dimension hyper cube with the side r, then its volume is:

$$V(C_d(r)) = V(C_d(1)) * r^d, \quad \text{here } V(C_d(1)) = 2^d \tag{2}$$

So the proportion of the volume of inscribed circle and cube decrease to zero as the dimension d increase.

$$\frac{V(B_d(1))}{V(C_d(1))} = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1) * 2^d} \xrightarrow{d \to \infty} 0 \tag{3}$$

From the perspective of the geometric, with the increase of dimension, the volume of hypercube will expand to the corners, and the center become more and more sparse.

For high dimensional problem, the area nearby the global optimal becomes largely sparse as the dimension increase, so the more difficult to search the global optimal. In simulation, we also found the SPSO algorithm might have difficulties in higher dimensions. For many of the problems more than 100 dimensions we looked at, PSO's predisposition towards premature convergence worsened as the dimensionality of the problem grew. The increased problem space, along with the sparse population of

particles within that problem space, often caused the swarm quickly converging on relatively poor solution.

3 Dynamic Dimension Crossover Based on PSO

Our initial goal was to examine the ability of PSO in high dimensional problem spaces, and see if there might be a simple way of improving its performance on those types of problems. To that end, our research was narrowed to the techniques-Crossover, which borrowed the concept from Genetic Algorithm (GA). We improve upon the crossover operator and put forward the concept "Dimension Crossover" that do not require any large or computationally expensive changes to the PSO algorithm.

3.1 Standard PSO

The PSO algorithm was first proposed by Kennedy and Eberhart as a new method for function optimization. The inspiration for PSO came from observing the social behavior found in nature in such things as a flock of birds, a swarm of bees, or a school of fish.

Shi and Eberhart found that the inertia weight w plays an important role for the convergence of PSO, then published the article "A Modified Particle Swarm Optimizer"in 1998, brought forward the following formulas at the first time:

$$V_t = w * V_{t-1} + c1 * rand * (pBest - p_t) + c2 * rand * (gBest - p_t) \quad (4)$$

$$p_{t+1} = V_t + p_t \quad (5)$$

The formulas are thought as standard PSO by most researchers. The weight is used to make a trade-off between the global and local exploration abilities of the swarm, since large value of w facilitate global exploration of the search space (visiting new regions) while small values facilitate local exploration. The common measure is to use large value of w at the first steps of the algorithm and gradually decrease it during the optimization in order to perform more refined search of the already detected promising regions. Now many improved algorithms are based upon Standard PSO.

3.2 Dimension Crossover

The PSO algorithm is modeled around the concept of swarming, and therefore does not make use of an explicit crossover operator. Actually, the process of particle flying is one special crossover that the current particle and the gBest generate child through crossover.

If we direct use the single-point stochastic crossover to get the better individual, simulation prove it invalid. In the simulation, we also found that a majority of dimensions of the particles are highly fine in the latter part of evolution, but only a minority of dimensions is poor. For example, in the process of optimizing 100 dimension sphere functions, one particle is (0.0011, 0.0023… 0.0002, 1.6588), another particle is (0.0012, 1.4001… 0.0002, 0.0018), how to get the offspring without the poor

dimensions become a critical problem. Here the good offspring means that it does not include the bad dimension such as 1.6588 and 1.4001. It was therefore we bring forward the concept of dimension crossover operator. By dimension crossover, we could get the offspring such as (0.0011, 0.0023 … 0.0002, 0.0018) which must be better than the father.

First we make an **assumption:** there are two particles A, B and their position, $A=a_1a_2\ldots a_d$, $B=b_1b_2\ldots b_d$, here assuming that a_i and b_j is poor value of the i and j dimension $(i \neq j)$ and the values of other dimension are fine for the optimization functions.

Definition 1. one mapping from father space to individual space: $T_c : S^2 \to S$ is called crossover operator, specially:

(1) If we stochastically make one gene as crossover point equiprobably, then all data beyond that point in either organism string is swapped between the two parent organisms. We take the first one of the resulting organisms as the child, here Tc is called One-point crossover (OPC).

(2) If we make One-point crossover with a fixed probability Pc, then we called the Tc One-point stochastic crossover (OPSC).

(3) If we swapped all the dimentions of the parents by the way One-point crossover, we call the Tc Dimension crossover (DC).

Theorem 1. Supposed that $(A,B)\in S^2, Y\in S, K\in N$, Y is the good offspring better than parents, and K is the sum of new offspring better than the two parents, Pc is crossover probability, then

$$P_{dc}\{T_c(A,B)=Y\} \geq P_{opc}\{T_c(A,B)=Y\} \geq P_{opsc}\{T_c(A,B)=Y\} \qquad (6)$$

That is to say, the probability of the dimension crossover producing better offspring is large than one-point crossover and one-point stochastic crossover.

Proof 1. if $A=a_1a_2\ldots a_d$, $B=b_1b_2\ldots b_d$, a_i and b_j $(i \neq j)$ is poor dimension, the crossover point is t, then:

$T_c(A,B)=a_1a_2\cdots a_tb_{t+1}\cdots b_d$, this child may be Y or not. When t=1, 2… d, there are K points could produce better offspring. So the probability of One-point crossover is:

$$P_{opc}\{T_c(A,B)=Y\}=\frac{K}{d-1}=\frac{|i-j|}{d-1}$$

Because dimension crossover make d-1 times One-point crossover, so the probability of dimension crossover is:

$$P_{dc}\{T_c(A,B)=Y\}=\frac{K}{d-1}(d-1)=K=|i-j|$$

$\because (i \neq j)$, so $P_{dc}\{T_c(A,B) = Y\} = 1$

For the one-point stochastic crossover operator, the probability is:

$$P_{opsc}\{T_c(A,B) = Y\} = \frac{KPc}{d-1} = \frac{|i-j|Pc}{d-1}$$

Obviously, $\frac{|i-j|Pc}{d-1} \leq \frac{|i-j|}{d-1} \leq 1$,

Thus, the theorem is proved:

$$P_{dc}\{T_c(A,B) = Y\} \geq P_{opc}\{T_c(A,B) = Y\} \geq P_{opsc}\{T_c(A,B) = Y\}$$

So we could conclude the dimension crossover has a great probability of producing good offspring.

Given d =10, Pc =0.7, a_2 and b_4 are poor values, the probability good offspring is:

$$P_{dc}\{T_c(A,B) = Y\} = 1, P_{opc}\{T_c(A,B) = Y\} = 0.2, P_{opsc}\{T_c(A,B) = Y\} = 0.14$$

3.3 Dynamic Dimension Crossover for High Dimensional Problems

In the end of the evolution, especially for the high-dimensional problems, the phenomenon of premature occurred easily. In this paper we proposed the dynamic dimension crossover PSO (DDC-PSO) to solve this problem, the main idea is: judge whether the swarm converged, if it converged, then use dimension crossover to change the location of particles, thereby produce better individuals. But there are two issues here; one is how to judge whether the swarm become premature and then begin dimension crossover. The other is how to select two particles from the population as the father, then produce 2 *(d-1) children by dimension crossover, and the children how to migrate to make up of next population and evolve on and on.

For the first issue, we compute all the particles' fitness, if the values do not varies for the consecutive ten generations, then we could make a presumption that dimension crossover operator should be employed to avoid premature convergence.

For the second issue, we choose the best individual and the worst individual as the parents. Actually, in high-dimensional space, the worst individual, sometimes in some ways is better than the best individual. For example, for the three-dimensional spherical function, the optimal value is (0, 0, 0), there are two particles which the positions (0.001, 6, 8) and (100, 0.001, 0.001), the fitness value of the former is better than the latter, but from the point of position, the former is much worse than the latter.

The last thing is to solve the migration of the 2 * (d-1) children produced by dimension crossover, we sort all the particles' fitness value; the following section will show the results as the different proportion of all the children migrates to the next population.

The flow chart of DDC-PSO is as follows:

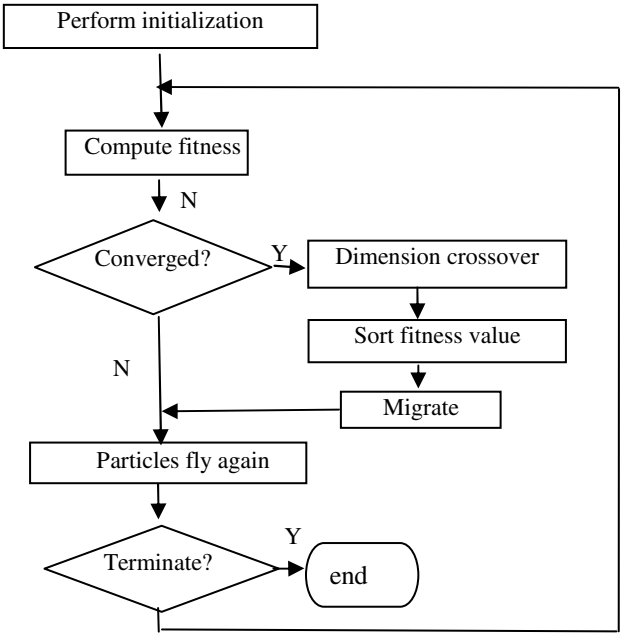


Fig. 1. The flow chart of DDC-PSO

4 Simulation Result and Discussion

4.1 Standard Test Functions and Experimental Settings

To demonstrate the effectiveness of proposed DDC-PSO, five well known test functions have been chosen to evaluate the performance between SPSO, CPSO and DDC-PSO. The five test functions are:

Sphere Function : $F1 = \sum_{i=1}^{N} x_i^2, x \in [-2, 2]$

Rastigrin function: $F2 = \sum_{n=1}^{N} [x_n^2 - 10\cos(2\pi x_n) + 10], x \in [-2, 2]$

Rosenbrock Function: $F3 = \sum_{n=1}^{N-1} [100(x_{n+1} - x_n^2)^2 + (x_n - 1)^2], x \in [-2, 2]$

Griewank Function: $F4 = 1 + \frac{1}{4000} \sum_{n=1}^{N} x_n^2 - \Pi_{n=1}^{N} \cos(x_n / \sqrt{n}), x \in [-2, 2]$

Ackley function:

$$F5 = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2}) - \exp(\frac{1}{N}\sum_{n=1}^{N} \cos(2\pi x_n)), x \in [-2, 2]$$

Three algorithms are implemented with the parameters shown in table 1. All the algorithms are set to terminate after 1000 fitness evaluations and the final result is taken as the average of 50 independent runs.

Table 1. Parameters used to implement three algorithms

	SPSO	CPSO	DDC-PSO
Dimension D	100	100	100
Population Size L	30	30	30
Learning factor C	2	2	2
Inertia weight Winial,Wend	0.9~0.4	0.9~0.4	0.9~0.4
Crossover operator		Discrete crossover	Dynamic dimension crossover
Max evaluations	1000	1000	1000

4.2 Results and Discussion

Figs.2-6 show the comparison of optimization performances for the five test functions, and Table 2 show the summary of our simulation results. It can be seen that DDC-PSO provides a much superior optimization performance than SPSO and CPSO.

Table 2. Results after 50 independent runs

	Algorithm	Avg	Std. Dev.
F1	SPSO	9.856692	7.6828
	CPSO	19.69296	7.7430
	DDCPSO	0.3451845	1.7575
F2	SPSO	1027.6582	72
	CPSO	1129.3264	73.5566
	DDCPSO	353.5827	25.3390
F3	SPSO	1655.070	2854.7
	CPSO	3403.090	3213.1
	DDCPSO	243.9244	629.7405
F4	SPSO	0.2248436	0.0825
	CPSO	0.3753790	0.0721
	DDCPSO	0.01440611	0.0808
F5	SPSO	3.009859	0.2291
	CPSO	3.424890	0.1977
	DDCPSO	0.8119493	0.0856

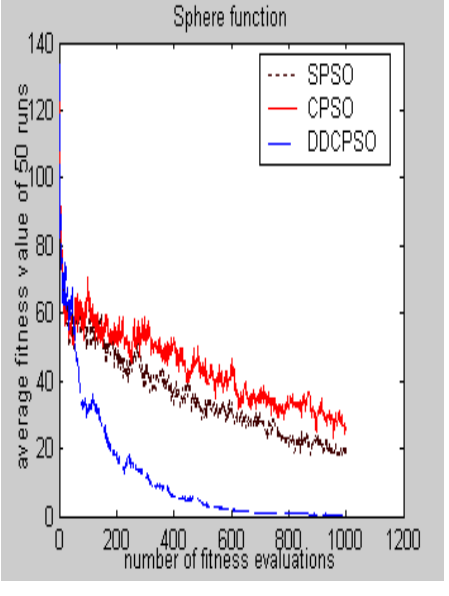


Fig. 2. Comparison of three algorithms for Sphere functions of dimension 100

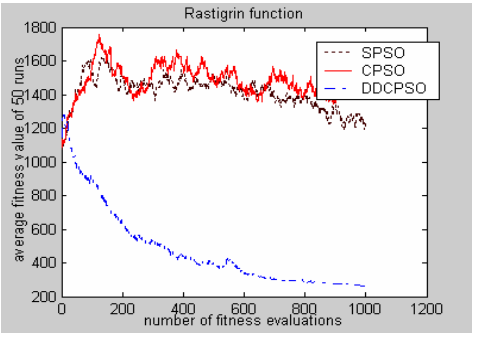


Fig. 3. Comparison of three algorithms for Rastigrin functions of dimension 100

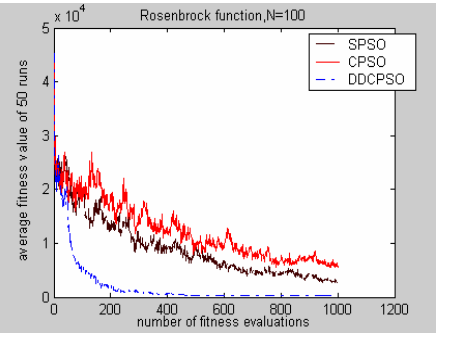


Fig. 4. Comparison of three algorithms for Rosenbrock functions of dimension 100

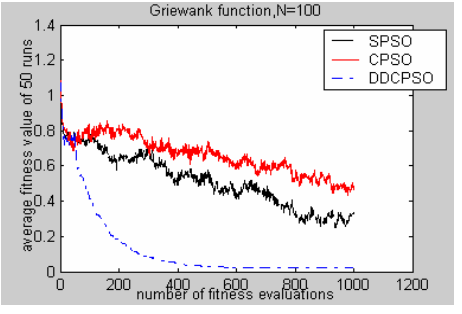


Fig. 5. Comparison of three algorithms for Griewank functions of dimension 100

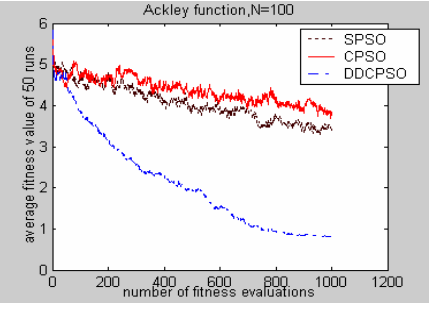


Fig. 6. Comparison of three algorithms for Ackley functions of dimension 100

Simulation are implemented on the platform of WinXP, MATLAB 6.5, and the hardware is Pentium4 2.4G, DDR2 512M.

In Fig 2-6, the black curve shows the performance of SPSO. It has difficulty in the optimization of high-dimensional problem. During the fitness evaluations the curve decline slowly and largely fluctuated, and for the Rastigrin function, the fitness start ups and downs and the population converged quickly. This show that SPSO has poor performance for high dimension problem and it is even impossible to optimize for some special high-dimensional functions.

The red curve shows the performance of CPSO. As one improved PSO, its ability in optimizing has not strengthened but weakened the performance of the PSO. Imagine that after the particles fly, then cross blindly, thus only makes the effect of particle flying greatly reduced, so simple discrete uniform crossover make the performance decline. Therefore, we proposed one new dimension crossover operator after analysis of the reasons for the function to optimize.

The blue curve shows the performance of DDCPSO. For all the test functions, ADCPSO has shown the perfect combination of the convergence speed and precision. From the point of compute complexity, for the three algorithms, DDCPSO would,

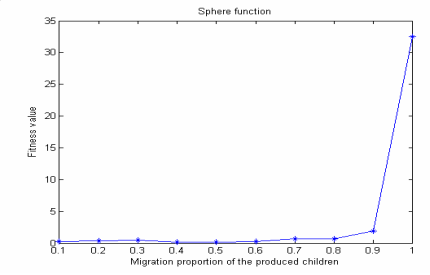


Fig. 7. Migration proportion of the produced children by DDC-PSO for the sphere function

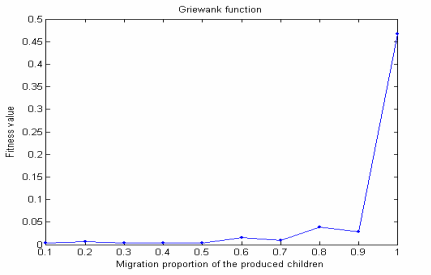


Fig. 8. Migration proportion of the produced children by DDC-PSO for the Griewank function

however, increases the computational cost in evaluating the fitness, the compute complexity of all the three algorithms are similar.

At last, we will show that different migration proportion of the produced children has a big effect on the performance of PSO for high dimensional problem.Fig.7 and Fig.8 shows a smaller migration proportion will advance the performance of the algorithm; a greater increase in the proportion of migration, the algorithm will lost the diversities and converge quickly. Therefore, the proposed ratio of migration is at 1% to 50%.

5 Conclusions and Future Work

In this paper, we have for the first time proposed a new algorithm called DDC-PSO for solving high dimensional optimization problems.

Our simulation results have shown that the proposed DDC-PSO is more suitable technique for solving high dimensional optimization problems than SPSO and CPSO. At the same time, we have studied the effect on the performance of the new algorithm under different population migration; finally we put forward reasonable migration ratio. We are also currently investigating the high dimensional problems with multiple constraints, and the theory is underway for further in-depth analysis.

References

[1] Eberhart, R., Kennedy, J.: New optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43. IEEE CS Press, Nagoya (1995)

[2] Lu, T.: The Theory and Application of the Dimension Reduction on the High-Dimensional Data Set, Science doctoral dissertation, National University of Defense Technology (2005)

[3] Potter, M.A., De Jong, K.A.: A cooperative co-evolutionary approach to function optimization. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994)

[4] Huanga, T., Ananda, S.M.: Micro-particle swarm optimizer for solving high dimensional optimization problems. Applied Mathematics and Computation, 1148–1154, October 15 (2006)
[5] Lei, K., Qiu, Y., He, Y.: An Effective Particle Swarm Optimizer for Solving Complex Functions with High Dimensions. Computer Science 33 (2006)
[6] Zhang, J., Hu, A., Guan, H.: A Combined SA-PSO Algorithm Based on PSO and SA Algorithms. Water Power 34(3) (2008)
[7] Gao, H., Xu, W.-b., Sun, J.: A Quantum-particle swarm algorithm for optimizing high-dimension functions. Journal of Computer Applications 27(12) (2007)
[8] Li, L., Li, H.-q.: Solving for complex functions with high dimensions based on hybrid particle swarm optimization. Journal of Computer Applications 27(7) (2007)

Stability Analysis of Discrete Hopfield Neural Networks with Weight Function Matrix

Jun Li[1], Yongfeng Diao[1], Jiali Mao[1], Ying Zhang[1], and Xing Yin[2]

[1] School of Computer Science, China West Normal University, Nanchong 637002, China
lj_1202@126.com
[2] School of Computer Science, Pan Zhi Hua University, Panzhihua 637000, China
silverst@sohu.com

Abstract. Most matrixes of Discrete Hopfield neural networks(DHNNs) and DHNNs with delay are constant matrixes. However, most weight matrixes of DHNNses are variable in many realistic systems. So, the weight matrix and the threshold vector with time factor are considered, and DHNNs with weight function matrix (DHNNWFM) is described. Moreover, the result that if weight function matrix and threshold function vector respectively converge to a constant matrix and a constant vector that the corresponding DHNNs is stable or the weight matrix function is a symmetric function matrix, then DHNNWFM is stable, is obtained by matrix analysis.

1 Introduction

Discrete Hopfield neural networks(DHNNs) is one of the famous neural networks with a wide range of applications, such as content addressable memory, pattern recognition, and combinatorial optimization[1-5]. With the research of DHNNs developing in theory and application, the model is more and more complex. It is well known that nonautonomous phenomena often occur in many realistic systems. Particularly when we consider a longterm dynamical behaviors of the system and consider seasonality of the changing environment, the parameters of the system usually will change with time[6]. But most matrixes of DHNNs and DHNNs with delay are constant matrixes[1-5,7-9]. This paper describes the model of discrete Hopfield neural networks with weight function matrix(DHNNWFM) that the weight changes with time. The application heavily depends on the networks' dynamical behaviors. Therefore, the analysis of dynamical behaviors has important leading significance in the design and application of Hopfield neural networks. In [6,10], the stability properties of these models for weights changed with time have been studied. However, the weight function matrix and the threshold function vector are all periodic functions. Based on this, this paper considers that the weight function matrix and the threshold function vector respectively converge to a constant matrix and a constant vector and the weight function matrix is a symmetric function matrix, and analyzes the stability of the model.

This paper has the following organization. In this section (Section 1), we provide the introduction. In Section 2, we introduce some basic concepts about the limit and the difference. In Section 3, we give a description of DHNNWFM. In Section 4, we

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 760–768, 2008.
© Springer-Verlag Berlin Heidelberg 2008

analyze the stability of DHNNWFM. The last section offers the conclusions of this paper.

2 Basic Concepts[11,12]

All When equidistant point t_k is equal to $t_0 + kh$ ($k = 1, 2, \cdots$), f_k is equal to $f(t_k)$ by $y = f(t)$. In here, step length h is a constant.

Definition 1. To discrete time factor t with step length h, if positive ε is arbitrary, natural number N exists, and when $k >$ N, ε-epsilon neighborhood set of definite vector X contains vectors $\{X(t_k)\}$, then $|X(t_k) \to X| < \varepsilon$. We say that X is a limit of vectors $\{X(t_k)\}$, record as: $\lim_{k\to\infty} X(t_k) \to X$.

Definition 2. To discrete time factor t with step length h, if relation between $a_{ij}(t_k)$ of $A(t_k) = (a_{ij}(t_k))_{m\times n}$ and a_{ij} of $A = (a_{ij})_{m\times n}$ satisfies $\lim_{k\to\infty} a_{ij}(t_k) \to a_{ij}$ ($i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$), then X is a limit of matrixes $\{A(t_k)\}$, record as: $\lim_{k\to\infty} A(t_k) \to A$.

Definition 3. $\Delta f_k = f_{k+1} - f_k$ is called difference of $f(t)$ at t_k with step length h.

Definition 4. Suppose $X(t) = (x_j(t))$ ($j = 1, 2, \cdots, n$) is a function vector, $\Delta X(t_k) = X(t_{k+1}) - X(t_k) = (\Delta x_j(t_k))$ ($j = 1, 2, \cdots, n$) is called vector difference at t_k with step length h.

Definition 5. Suppose $A(t) = (a_{ij}(t))_{m\times n}$ ($i = 1, 2, \cdots, m; j = 1, 2, \cdots, n$) is a function matrix, $\Delta A(t_k) = A(t_{k+1}) - A(t_k) = (\Delta a_{ij}(t_k))$ ($i = 1, 2, \cdots, m; j = 1, 2, \cdots, n$) is called matrix difference at t_k with step length h.

3 Discrete Hopfield Neural Networks with Weight Function Matrix Model

Discrete Hopfield neural networks(DHNNs) has been described. DHNNs with weight function matrix (DHNNWFM) varies with the discrete time factor t by step length h (in this paper, $h = 1$). Formally, let $N(t) = (W(t), \theta(t))$ be a DHNNs with n

neurons, which have the discrete time factor t with step length 1 and are denoted by $\{x_1, x_2, \cdots, x_n\}$. In the pair $(W(t), \theta(t))$, $W(t) = (w_{ij}(t))_{n\times n}$ is an $n \times n$ function matrix where $w_{ij}(t)$, which changes with time t, represents the connection weight from x_i to x_j, and $\theta(t) = (\theta_i(t))_{n\times 1}$ is an n-dimensional function vector where $\theta_i(t)$, which changes with time t, represents the threshold attached to the neuron x_i. The state of the i neuron at time t is denoted by $x_i(t)$. Each neuron is assumed to have two possible states: 1 and 0. The state of the network at time t is the vector $X(t) = (x_1(t), x_2(t), \cdots, x_n(t))^T \in R^n$. In general, the state, $X(t+1)$, of the network at time $t+1$ is a function of $\{W(t), \theta(t)\}$ and the state, $X(t)$, of the network at time t. The network is, thus, completely determined by the parameters $\{W(t), \theta(t)\}$, the initial value, $X(0)$, of the states, and the manner in which the neurons are updated(evolved). If at time step t, a neuron x_i, is chosen to be updated, then at the next step.

$$x_j(t+1) = \operatorname{sgn}(y_j(t)) = \begin{cases} 1, y_j(t) \geq 0, \\ -1, y_j(t) < 0. \end{cases}$$

where $y_i(t) = \sum_{j=1}^{n} w_{ij}(t) x_j(t) - \theta_i(t), i = 1, 2, \cdots n$

(1) The network is updated asynchronously, that is, only one neuron i is selected at time $t+1$. The updating rule is

$$x_j(t+1) = \begin{cases} \operatorname{sgn}(y_j(t)), j = i, \\ x_j(t), j \neq i. \end{cases}$$

(2) The network is updated synchronously, that is, every neuron i is selected at time $t+1$. The updating rule is

$$x_i(t+1) = \operatorname{sgn}(y_i(t)), i = 1, 2, \cdots, n$$

Let $N(t) = (W(t), \theta(t))$ be a DHNNWFM. If the output of n neurons does not change any longer after a limited time interval t from its initial state $X(0)$, that is $X(t+1) = X(t)$. Then we can say that the network is a stable state, and we call $X(t+1) = X(t)$ a stable point of N. In addition, we say that the network

converges about the initial state $X(t_0)$. It is easy to known if X is a stable point of DHNNWFM, then $X(t)=\mathrm{sgn}(W(t)X(t)-\theta(t))$. Sometimes we call X that satisfies the above formula a stable attraction factor of the network N. If X is an attraction factor DHNNWFM $N(t)=(W(t),\theta(t))$, we denote attraction domain of X as $\Gamma(X)$, which represents the set which consists of all of the initial state $X(0)$ attracted to X.

Let $N(t)=(W(t),\theta(t))$ be a DHNNWFM. $X(0),X(1),\cdots,X(r-1),(r\geq 1)$ are n-dimensional vectors. If $X(t+1)=\mathrm{sgn}(W(t)X(t)-\theta(t))$, $t=0,1,2,\cdots,r-2$ and $X(0)=\mathrm{sgn}(W(r-1)X(r-1)-\theta(r-1))$, then we called a limit cycle attraction factor of $N(t)$, sometimes abbreviated to limit cycle; its length is r, and is denoted by $(X(0),X(1),\cdots,X(r-1))$. Similar to the attraction domain of a stable attraction factor, the attraction domain of cycle attraction factor $(X(0),X(1),\cdots,X(r-1))$, denoted by $\Gamma(X(0),X(1),\cdots,X(r-1))$, represents the set that consists of all the possible initial states which are attracted to $(X(0),X(1),\cdots,X(r-1))$.

4 Stability Analysis of DHNNWFM

Theorem 1. Let $N(t)=(W(t),\theta(t))$ be a DHNNWFM. $W(t)=(w_{ij}(t))_{n\times n}$ is an $n\times n$ function matrix. $\theta(t)=(\theta_1(t),\theta_2(t),\cdots,\theta_n(t))^T\in R^n$ is an n-dimensional function vector.

1) If DHNNs $N=(W,\theta)$ is stable, $\lim_{t\to\infty}W(t)\to W$ and $\lim_{t\to\infty}\theta(t)\to\theta$, then $N(t)$ is stable too.

2) If $\Delta W(t)>0\wedge\Delta\theta(t)<0$, then any initial state X of $N(t)$, but $(0,0,\cdots,0)_{n\times 1}$, is attracted to $(1,1,\cdots,1)_{n\times 1}$.

Proof: part one

Because of $\lim_{t\to\infty}W(t)\to W$, when $t>\mathsf{N}$, we have $|W(t)\to W|<\varepsilon_1$

Because of $\lim_{t\to\infty}\theta(t)\to\theta$, when $t>\mathsf{N}$, we have $|\theta(t)\to\theta|<\varepsilon_2$.

So, when $t > \mathrm{N}$, $\left|\theta(t) \to \theta\right| < \varepsilon \wedge \left|W(t) \to W\right| < \varepsilon$, $\varepsilon = \max\{\varepsilon_1, \varepsilon_2\}$. Then $\theta - \varepsilon \mathrm{H} < \theta(t) < \theta + \varepsilon \mathrm{H}$ ($\mathrm{H} = (h_i = 1)_{n\times 1}, 1 \le i \le n$) and $W - \varepsilon \mathrm{E} < W(t) < W + \varepsilon \mathrm{E}$ ($\mathrm{E} = (e_{ij} = 1)_{ij}, 1 \le i, j \le n$).

The energy function set before time t: $E_0 = \{E(0), E(1), \cdots, E(t)\}$, E_0 does not affect the later energy function.

When $t > \mathrm{N}$, the energy function is

$$E_t(t) = -\tfrac{1}{2} X^T(t) W(t) X(t) - X^T(t)\theta(t)$$

Let $L(t)$ be $\frac{1}{2} X^T(t)\mathrm{E}X(t) + X^T(t)\mathrm{H}$.

$$E(t) - \varepsilon L(t) \le E_t(t) \le E(t) + \varepsilon L(t),$$

$$\Delta E(t) - \varepsilon \Delta L(t) \le \Delta E_t(t) \le \Delta E(t) + \varepsilon \Delta L(t)$$

$$\max\{\Delta L(t) \mid t = 0, 1, 2, \cdots\} = n^2/2 + n.$$

If $\varepsilon \gg \dfrac{1}{n^2/2 + n}$ or $\varepsilon = \dfrac{1}{t}$, then $\lim\limits_{t\to\infty} \Delta E_t(t) = \Delta E(t)$

Because DHNNs $N = (W, \theta)$ is stable, $\Delta E(t) \le 0$.

So, when $\varepsilon \gg \dfrac{1}{n^2/2 + n}$, $\Delta E_t(t) \le 0$ that $N(t)$ is stable.

Part two

If $\Delta W(t) > 0$, then $W(t)$ is monotonically accelerating .

If $\Delta\theta(t) < 0$, then $\theta(t)$ is monotonically decelerating.

So, $\exists$ time l, L_1 is greater than L_2.

$L_1 = \min\{w_{ij}(l) \mid 1 \le i, j \le n\}$ and $L_2 = \max\{\theta_i(l) \mid 1 \le i \le n\}$. If $L_1 > L_2$, any initial state X , which can activate only one neuron, is attracted to $(1, 1, \cdots, 1)_{n\times 1}$. But $(0, 0, \cdots, 0)_{n\times 1}$ can not activate any neuron. This completes the proof.

Theorem 2. Let $N(t) = (W(t), \theta(t))$ be a DHNNWFM. $W(t) = (w_{ij}(t))_{n\times n}$ is an $n \times n$ function matrix. $\theta(t) = (\theta_1(t), \theta_2(t), \cdots, \theta_n(t))^T \in R^n$ is an n-dimensional function vector.

1) If $W(t)$ is a symmetric function matrix, then $N(t)$ is stable;

2) If $\Delta w_{ii}(t) \geq \sum_{j=1,\neq i}^{n} \left(\left| \Delta w_{ij}(t) \right| + 2 \left| w_{ij}(t) - w_{ji}(t) \right| \right) \wedge \theta_i(t) = 0,$

$i = 1, 2, \cdots, n$, then $N(t)$ is stable.

Proof: part one

Consider energy function (Lyapunov function) of the DHNNWFM as follows:

$$E(t) = -\tfrac{1}{2} X^T(t) W(t) X(t) - X^T(t) \theta(t)$$

$$= -\tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(t) x_i x_j + \sum_{j=1}^{n} \theta_j(t) x_j$$

$$= \sum_{j}^{n} \left\{ -\tfrac{1}{2} \sum_{i=1}^{n} w_{ij}(t) x_i x_j + \theta_j(t) x_j \right\}$$

We define energy function (Lyapunov function) of the neuron j as follows:

$$E^j(t) = -\tfrac{1}{2} \sum_{i=1}^{n} w_{ij}(t) x_i x_j + \theta_j(t) x_j$$

So, $E(t) = \sum_{j=1}^{n} E^j(t)$.

$$\Delta E^j(t) = \frac{\partial E}{\partial x_j} \Delta x_j = \frac{\partial E^j}{\partial x_j} \Delta x_j$$

$$= \left[\left(-\tfrac{1}{2} \right) \sum_{i=1}^{n} \left(w_{ij}(t) x_i \frac{x_j}{\partial x_j} + w_{ji}(t) \frac{x_j}{\partial x_j} x_i \right) + \theta_i(t) \frac{x_j}{\partial x_j} \right] \Delta x_j$$

If $w_{ij}(t) = w_{ji}(t)$ and $w_{ii}(t) = 0$, then

$$= -\left(\sum_{\substack{i=1 \\ i \neq j}}^{n} w_{ij}(t) x_i - \theta_i(t) \right) \Delta x_j$$

Suppose $U_j = \sum_{i=1,\neq j}^{n} w_{ij}(t) x_i$. If $U_j \geq \theta_j$, then $\Delta x_j \geq 0$, so, $\Delta E^j(t) \leq 0$; If $U_j \leq \theta_j$, then $\Delta x_j \leq 0$, so, $\Delta E^j(t) \leq 0$.

In any case, $E(t) \leq 0$ holds. So, $N(t) = \left(W(t), \theta(t) \right)$ is stable.

part two

Consider energy function (Lyapunov function) of the DHNNWFM as follows:

$$E(t) = -\tfrac{1}{2} X^T(t) W(t) X(t) - X^T(t) \theta(t) = -\tfrac{1}{2} X^T(t) W(t) X(t)$$

So, $\Delta E(t) = E(t+1) - E(t)$

$= -\frac{1}{2} X^T(t+1) W(t+1) X(t+1) + \frac{1}{2} X^T(t+1) W(t) X(t+1)$

$-\frac{1}{2} X^T(t+1) W(t) X(t+1) + \frac{1}{2} X^T(t) W(t) X(t+1) +$

$\frac{1}{2} \Delta X^T(t) W(t) X(t) - \frac{1}{2} X^T(t) W(t) \Delta X(t)$

$-\frac{1}{2} X^T(t) W(t) X(t+1) + \frac{1}{2} X^T(t) W(t) X(t)$

$= -\frac{1}{2} X^T(t+1) \Delta W(t) X(t+1) - \frac{1}{2} X^T(t) \left(W(t) + W^T(t)\right) \Delta X(t)$

$-\frac{1}{2} \Delta X^T(t) W(t) \Delta X(t)$

If $\Delta X(t) = 0$, by $\Delta w_{ii}(t) \geq \sum_{j=1,\neq i}^{n} \left(\left| \Delta w_{ij}(t) \right| + 2 \left| w_{ij}(t) - w_{ji}(t) \right| \right)$, we have:

$$\Delta E(t) = -\frac{1}{2} X^T(t+1) \Delta W(t) X(t+1) \leq -\frac{1}{2} \sum_{i=1}^{n} \left(\Delta w_{ii}(t) - \sum_{j=1}^{n} \left| \Delta w_{ij}(t) \right| \right) \leq 0.$$

If $\Delta X(t) \neq 0$, then $\Delta X(t) = -2X(t) = 2X(t+1)$,

$\Delta E(t) = -\frac{1}{2} X^T(t) \left(\Delta W(t) + 2W(t) - 2W^T(t) \right) X(t)$

By $\Delta w_{ii}(t) \geq \sum_{j=1,\neq i}^{n} \left(\left| \Delta w_{ij}(t) \right| + 2 \left| w_{ij}(t) - w_{ji}(t) \right| \right)$, we have:

$$\Delta E(t) \leq -\frac{1}{2} \sum_{i=1}^{n} \left(\Delta w_{ii}(t) - \sum_{j=1,\neq i}^{n} \left| \Delta w_{ij}(t) + 2 w_{ij}(t) - 2 w_{ji}(t) \right| \right) \leq 0$$

Then, $\Delta E(t) \leq 0$ .This completes the proof.

Corollary 1. Let $N(t) = \left(W(t), \theta(t) \right)$ be a DHNNFWM. If $\Delta w_{ii}(t) \geq \sum_{j=1,\neq i}^{n} \left(\left| \Delta w_{ij}(t) \right| + 2 \left| w_{ij}(t) \right| + 2 \left| w_{ji}(t) \right| \right) \wedge \theta_i(t) = 0, i = 1, 2, \cdots, n$, then $N(t)$ is stable.

Proof: Consider energy function (Lyapunov function) of the DHNNWFM as follows:

$E(t) = -\frac{1}{2} X^T(t) W(t) X(t)$. So, $\Delta E(t) = E(t+1) - E(t)$

$= -\frac{1}{2} X^T(t+1) \Delta W(t) X(t+1) - \frac{1}{2} X^T(t) \left(W(t) + W^T(t) \right) \Delta X(t)$

$-\frac{1}{2} \Delta X^T(t) W(t) \Delta X(t)$

If $\Delta X(t)=0$, by $\Delta w_{ii}(t)\geq \sum_{j=1,\neq i}^{n}\left(\left|\Delta w_{ij}(t)\right|+2\left|w_{ij}(t)\right|+2\left|w_{ji}(t)\right|\right)$, we have:

$$\Delta E(t)=-\tfrac{1}{2}X^{T}(t+1)\Delta W(t)X(t+1)\leq-\tfrac{1}{2}\sum_{i=1}^{n}\left(\Delta w_{ii}(t)-\sum_{j=1}^{n}\left|\Delta w_{ij}(t)\right|\right)\leq 0.$$

If $\Delta X(t)\neq 0$, then $\Delta X(t)=-2X(t)=2X(t+1)$,

$$\Delta E(t)=-\tfrac{1}{2}X^{T}(t)\left(\Delta W(t)+2W(t)-2W^{T}(t)\right)X(t)$$

By $\Delta w_{ii}(t)\geq \sum_{j=1,\neq i}^{n}\left(\left|\Delta w_{ij}(t)\right|+2\left|w_{ij}(t)\right|+2\left|w_{ji}(t)\right|\right)$, we have:

$$\Delta E(t)\leq-\tfrac{1}{2}\sum_{i=1}^{n}\left(\Delta w_{ii}(t)-\sum_{j=1,\neq i}^{n}\left|\Delta w_{ij}(t)+2w_{ij}(t)-2w_{ji}(t)\right|\right)\leq 0$$

Then, $\Delta E(t)\leq 0$ .This completes the proof.

5 Conclusion

In this paper, we introduce the DHNNs with weight function matrix (DHNNWFM). We mainly analyze the stability of DHNNWFM with matrix analysis and obtain that if weight function matrix and threshold function vector respectively converge to a constant matrix and a constant vector that the corresponding DHNNs is stable or the weight function matrix is a symmetric function matrix, then DHNNWFM is stable.

Acknowledgement

This work was supported by the Important Project Function of Science and Technology Bureau of Sichuan Province of China (Grant No. 05JY029-093).

References

[1] Hopfield, J.J.: Neural networks and physical systems emergent collective computational abilities. Proc. Nat. Acad. Sci. USA 79, 2554–2558 (1982)
[2] Bruck, J., Goodman, J.W.: A generalized convergence theorem for neural networks. IEEE Trans. on Inform Theory 34, 1089–1092 (1988)
[3] Goles, E., Fogelman, F., Pellegrin, D.: Decreasing energy functions as a tool for studying threshold Network. Discrete Applied Mathematics 12, 261–277 (1985)
[4] Xu, Z.B., Kwong, C.P.: Global Convergence and Asymptotic Stability of Asymmetric Hopfield Neural Networks. J. Mathematical Analysis and Applications 191, 405–427 (1995)
[5] Lee, D.L.: New stability conditions for Hopfield neural networks in partial simultaneous update mode. IEEE Trans. on Neural Networks 10, 975–978 (1999)

[6] Xiang, H., Yan, K.M., Wang, B.Y.: Existence and global exponential stability of periodic solution for delayed high-order Hopfield-type neural networks. Physics Letters A 352, 341–349 (2006)
[7] Zhang, Q., Wei, X., Xu, J.: On Global Exponential Stability of Discrete-Time Hopfield Neural Networks with Variable Delays. Discrete Dynamics in Nature and Society. Article ID: 67675, 9p (2007)
[8] Xu, B., Liu, X., Liao, X.: Global asymptotic stability of high-order Hopfield type neural networks with time delays. Comput. Math. Appl. 45(10-11), 1729–1737 (2003)
[9] Kosmatopoulos, E.B., Polycarpou, M.M., Christodoulou, M.A., Ioannou, P.A.: High-order neural network structures for identification of dynamical systems. IEEE Trans. Neural Networks 6(2), 422–431 (1995)
[10] Xiang, H., Yan, K.M., Wang, B.Y.: Existence and global exponential stability of periodic solution for delayed discrete high-order Hopfield-type neural networks. Discrete Dynamics in Nature and Society 3, 281–297 (2005)
[11] Bellman, R.: Introduction to Matrix analysis. The Rand Corporation (1970)
[12] Stoer, J., Bulirsch, R.: Introduction to Numerical analysis. Springer, New York (1980)

Hybrid Ant Colony Algorithm and Its Application on Function Optimization*

Bo Liu, Huiguang Li, Tihua Wu, and Qingbin Zhang

Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
q_water2003@yahoo.com.cn

Abstract. A new hybrid ant colony algorithm was proposed. Firstly, weight factor was introduced to the binary ant colony algorithm, and then we obtained a new probability by combining probability model of Population based incremental learning (PBIL) with transfer probability of ants pheromone . The new population are produced by probability model of PBIL, transfer probability of ants pheromone and the probability of proposed algorithm so that population polymorphism is ensured and the optimal convergence rate and the ability of breaking away from the local minima are improved. Optimization simulation results based on the benchmark test functions show that the hybrid algorithm has higher convergence rate and stability than binary ant colony algorithm (BACA) and Population based incremental learning (PBIL).

Keywords: Binary ant colony algorithm (BACA), function optimization,Population based incremental learning (PBIL), probability model.

1 Introduction

Ant Colony Optimization (ACO) is a stochastic meta-heuristic for solutions to combinatorial optimization problems. Since its first introduction by M. Dorigo and his colleagues [2] in 1991, ACO has been successfully applied to a wide set of different hard combinatorial optimization problems[1, 3], such as traveling salesman problems, quadratic assignment problems, and vehicle routing problems. The main idea of ACO is the cooperation of a number of artificial ants via pheromone laid on the path. Each ant contributes a little effort to the solution, while the final result is an emergent result of the ants's interactions.

Although ACO has been proved to be one of the best meta-heuristics in some combinatorial optimization problems [1, 3], However, it still has some shortcomings in the continuous domain , for example: at initial stages it lack pheromone, search slowly, and easily to trap into the local optimal solution if the quantity is large, which is to say premature convergence. So some researchers explorer combining the ant colony algorithm (ACA) with other intelligent algorithm to solve the problem, such as genetic algorithm,a certain extent,this kind of measure solve the question of local optimal solution, but it need to be researched more and more.Using for this idea, we considered combining the ant colony algorithm (ACA) with estimation of distribution algorithms (EDAs),because EDAs are Evolutionary Algorithms based on the

* This work is supported by NSF of Hebei Province #F2008001166 to Liubo.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 769–777, 2008.
© Springer-Verlag Berlin Heidelberg 2008

model probability what resemble to ant colony algorithm (ACA), Model probability of EDAs could lead the distributing of ants pheromone effectively, at the same time, the distributing of ants pheromone could inspire the foundation of model probability in EDAs.So we obtained a new group of ants by the new probability model and ensured the polymorphism of group,consequently we solved premature problem.

2 Binary Ant Colony Evolutionary

In the continuous domain ,the function optimization problem can be described as a triple (*D, F, f*), where *D* is the set of solutions, *F* is the set of feasible solutions, and *f* is the objective function. A solution x^* is called a global optimal solution if $f(x^*) = \min\{f(x) | x \in F\}$. Here we only discuss the minimization problems, since any maximization problem can be easily transferred to the relative minimization problem.

Define a directed graph G=(C, L), where the node set C is:

$$\{c_0(v_s), c_1(v_N^0), c_2(v_N^1), c_3(v_{N-1}^0), c_4(v_{N-1}^1), \ldots, c_{2N-3}(v_2^0), c_{2N-2}(v_2^1), c_{2N-1}(v_1^0), c_{2N}(v_1^1)\}$$

where v_s is a starting node, v_j^0 and v_j^1 are separately used to represent the value 0 or 1 of position b_j in binary system. To every node (j = 2,3,...N,), there are two directed arcs pointing to v_{j-1}^0 and v_{j-1}^1.

The inspiring source of ACO is the foraging behavior of real ants. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates it and carries some food back to the nest. During the return trip, the ant deposits a pheromone trail on the ground. The pheromone deposited, the amount of which may depend on the quantity and quality of the food, guides other ants to the food source. As it has been shown,indirect communication among ants via pheromone trails enables them to find shortest paths between their nest and food sources. This capability of real ant colonies has inspired the definition of artificial ant colonies that can find approximate solutions to hard combinatorial optimization problems.The central component of ACO algorithms is the pheromone model, which is used to probabilistically sample the search space.

$$P_{ij}^k = \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s \in allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} \tag{1}$$

Where P_{ij}^k is the probability of ant *k*'s shift from position i to position j at k time, α is the relative importance of track, β is the relative importance of visibility (β>=0), τ_{ij} is the remaining information on link i,j at t time, η_{ij} is the visibility of arc (i,j), allowed k = {0,1} represents the states that ant k is allowed to choose. Thanks to binary encoding, memory function is not needed; the ant can choose one edge according to the remaining quantity of pheromone on the edge in front. Besides, as time elapses, the former remaining information will disappear, After time period's n, ant completes one circle, and information on every routine will adjust as follows:

$$\tau_{ij}(t+n) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

Whereρ represents the durability of the track and $\rho \subset [0,\ 1]$, parameter 1-ρ represents the degree of information evaporation. Where $\Delta\tau_{ij}$ represents the remaining information left by ant k in routine ij during this circle. $\Delta\tau_{ij} = \dfrac{Q}{y_{sbest}}$,Where Q is a constant representing the number of tracks the ant left, y_{sbest} is the best solution currently.

In this paper,we renew the pheromone based on the ants weight factor of the different position in order to using the information of excellent ants effectively. Eventually, we used the good information of population effectively and avioded the mistake interference of no good population. *Ant_weight(1,n)* represents the matrix of weight factor:

$$Ant_weight(j) = \frac{\sum_{i=1}^{j} i}{\sum_{i=1}^{n} i} \quad \text{j=1,2 …n, } j \text{ represents the number of ants} \tag{3}$$

In case of *n=100,* the relation between weighting factor and ant position is presented in Fig.1:

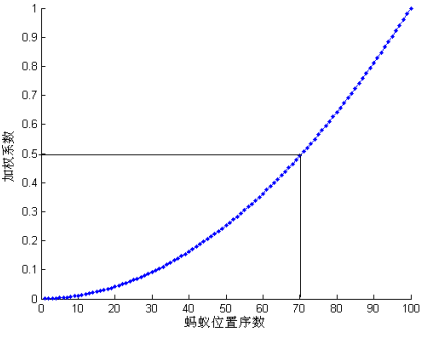


Fig. 1. Relation between weighting factor and ant position

3 PBIL Algorithm

PBIL algorithm is one of Estimation of Distribution Algorithms (EDAs) . In PBIL, the population of individuals is represented by a probability vector:

$$p_l x \ .p_l(x_1),p_l(x_2),.....p_l(x_n)) \tag{4}$$

Where $p_l(x_i)$ refers to the probability of obtain a value 1 in the i^{th} gene position in the l^{th} generation.

The algorithm begins by initializing the probability vector $p_0(x)$ with all elements set to 0.5. This means when sampling from this probability vector random solutions are created because the probability of generating a 1 or 0 is equal. As search progresses, the values in the probability vector are gradually learned towards values representing high evaluation solutions.

The evolution process in PBIL is accomplished as follows: At each generation, M solutions are generated based upon the probabilities specified in the current probability vector $p_l(x)$. The solutions are then evaluated and N best solutions (N•M) are selected.We denote them by: $x^l_{1:M}$ ....., $x^l_{i:M}$ ....., $x^l_{n:M}$.

These selected best solutions are used to update the probability vector by using a Hebbian inspired rule:

$$p_{l+1}(x) = (1-a)p_{lx} + a\frac{1}{N}\sum_{k=1}^{N} x^l_{k:M} \quad \text{Where } a\text{•}0,1 \text{ is the learning rate.} \tag{5}$$

After the probability vector is updated, a new set of solutions are generated by sampling from the new probability vector $p_{l+1}(x)$, and the cycle is repeated until some termination condition is satisfied, e.g., the probability vector is converged to either 0.0 or 1.0 for each bit position. The pseudo-code of PBIL Algorithm is shown in Fig.2.

PBIL

Initialize probability vector $p_0(x)$

While no convergence do

Begin

Using $p_i(x)$ obtain M solutions: $x^i_1,.....,x^i_M$

Evaluate and rank $x^i_1,.....,x^i_M$

Select the $N(N \le M)$ best solutions: $x^l_{1:M},.....x^l_{N:M}$

Update the probability vector $p_{i-1}(x) = (p_{i-1}(x_1).....p_{i-1}(x_n))$:

For i=1.....n do

$p_{i-1}(x_i) = (1-a)p_i(x_i) + a\frac{1}{N}\sum_{K=1}^{N} x^l_{i,k:M}$

End

Fig. 2. The pseudo-code of PBIL algorithm

4 The proposed Algorithm

In this section, we proposed a new hybrid ant colony algorithm ,named Estimation of Distribution ACA （EDACA）. The EDACA algorithm introduce probability model of EDAs into the binary ACA and coach the selection of ants pheromone.

EDACA adopt that combining the probability of EDAs with the transfer probability of ACA:

$$P_5 = （1-\beta） \times P_3 + \beta \times P_4 \quad (6)$$

P_3 represented the transfer probability of ACA,P_4represented the probability of EDAs,P_5 represented the combining probability and β means the combing factor, if β is too big,the algorithm would rely on probability model much more and trap into the local optimal solution easily, if β is too small, the effect of probability model would become very small. In order to keeping it's balance,we get β=0.5.

In order to ensuring polymorphism of population,we produce the new population by three methods:

1) 1/4 of swarms produced by P_3;
2) 1/4 of swarms produced by P_4;
3) 2/4 of swarms produced by P_5

The pseudo-code of EDACA Algorithm is shown in Fig.3.

```
EDACA
  Initailize the path of Ant_pop(n,m), probability and pheromone
vector
  while ((NC<=NC_max)&(besty>0.00001))
  paratau=1/(1+2.71828^(-10*(NC/NC_max-0.5)))
    Update the pheromone trail and probability model
    solution P3,P4,P5
 for i=1:round(adjnum*m)
      P5= （1－β） ×P3+β×P4
      obtain new swarms
  END
```

Fig. 3. Pseudo-code of PBIL Algorithm

5 Computational Results and Analysis

We selected 7 benchmark test functions to examine the proposed algorithm and compared with other optimum algorithms,such as Binary Ant Colony Evolutionary and Estimation of Distribution Algorithms.

Table 1. The benchmark test functions

Name	Function formula	Dim	Range	Optimal	Goal
Schaffers	$f(x) = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	2	[-100, 100]	0	10^{-5}
Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	[-10, 10]	0	10^{-10}
Sumsquares	$f(x) = \sum_{i=1}^{n} i \times x_i^2$	30	[-10, 10]	0	10^{-5}
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	30	[-100, 100]	0	10^{-5}
Rastrgrin	$f(x) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$	30	[-5.12, 5.12]	0	10^{-5}
Grievank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n}(x_i)^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600, 600]	0	10^{-5}
Ackly	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[-30, 30]	0	10^{-5}

The optimization for these functions is evaluation of minimum.We repeat 20 operations using ACA, PBIL and EDACA about 7 test functions, the results of maximum, minimum, average, generations and times were listed in table 2. The parameters of algorithms were all follows:

BACA:n（number of ants）=50，ρ=0.1，b%=50%,coding length for every variable is10;PBIL:M（number of population）=50，N=0.3M，α(learning factor)=0.1，coding length for every variable is 10;EDACA:n（number of ants）=50，ρ=0.1，b%=50%，N=0.3n，α=0.1，β=0.5, coding length for every variable is 10.

From tests, we can see that BACA and PBIL run into local minimum easily, the speed of computing is really quick and the solution is stable in EDACA.

Fig.4 is the effect of three algorithms based on *Griewank* function, it is very clearly.

Table 2. Optimization results comparison between BACA, PBIL and EDACA

Function	Algorithm	Max.	Min.	Average	Generations	Times (s)
Schaffers	BACA	0.0098	0	0.0069	362.50	1.3703
	PBIL	0.0100	0	0.0069	357.10	1.5894
	EDACA	0	0	0	1.39	0.0102
Matyas	BACA	1.5259e-5	0	2.2888e-6	102.75	0.4070
	PBIL	0	0	0	26.40	0.1359
	EDACA	0	0	0	1.82	0.0109
Sumsquares	BACA	3.7506	0.1083	1.1235	500	22.1656
	PBIL	81.4709	13.7223	48.4385	500	28.3702
	EDACA	0	0	0	43.80	3.7187
Sphere	BACA	14.6103	0.3433	5.4817	500	22.4079
	PBIL	938.1104	68.9697	368.1297	500	29.3031
	EDACA	0	0	0	42.75	3.6485
Rastrigin	BACA	33.4010	16.2315	23.2771	500	25.7368
	PBIL	71.1202	41.4762	52.2730	500	31.8156
	EDACA	0	0	0	36.05	3.0555
Griewank	BACA	1.1448	0.2442	0.6012	500	22.7274
	PBIL	5.4072	1.8099	3.3222	500	28.6906
	EDACA	0	0	0	45.90	3.8914
Ackly	BACA	2.8366	0.1096	0.6882	500	23.0290
	PBIL	7.2591	2.3905	4.3567	500	28.3203
	EDACA	8.8818e-16	0	1.6758e-16	41.35	3.6467

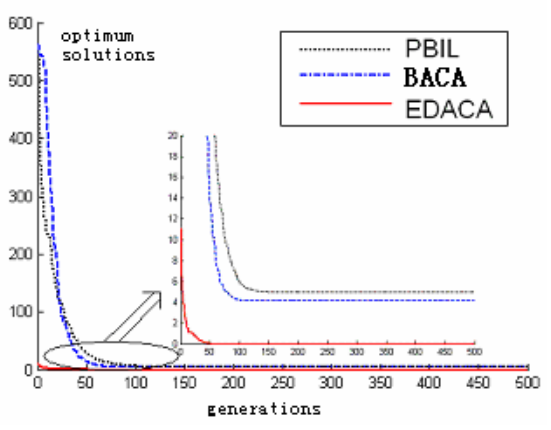


Fig. 4. Optimization results comparison of Griewank function between BACA, PBIL and EDACA

6 Conclusions

In this paper, we proposed a new hybrid ant colony algorithm that ants choose the route by the guidance of model probability of Population based incremental learning (PBIL) and control measure of pheromone emanating by weight factor, the probability model of three algorithms were syncretized to produce the new swarm of ants in order to ensure population polymorphism. It can be seen from the experimental results,that convergence speed can be improved with a suitable number of the selected solutions and a relatively large learning rate in proposed algorithm. Experimental results show that the proposed algorithm can converges more quickly and it can find more correct solutions for test functions than that of the standard PBIL and Binary Ant Colony Evolutionary.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems. Oxford University Press, Oxford (1999)
2. Dorigo, M., Maniezzo, V., Colorni, A.: Positive feedback as a search strategy. Technical Report 91-016, Dipartimento Di Elettronicaq – Politecnico Di Milano (1991)
3. Dorigo, M., Stützle, T.: Ant colony optimization. The MIT Press, Cambridge (2004)
4. Xiong, W., Wang, L., Yan, C.: Binary Ant Colony Evolutionary Algorithm. International Journal of Information Technology 12(3) (2006)
5. Petrovski, A., Shakya, S., McCall, J.: Optimising cancer chemotherapy using an estimation of distribution algorithm and genetic algorithms. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, pp. 413–418. ACM Press, New York (2006)
6. Yuan, B., Gallagher, M.: Playing in Continuous Spaces: Some Analysis and Extension of Population-Based Incremental Learning. In: Proceedings of Congress of Evolutionary Computation (CEC), pp. 443–450 (2003)

7. Shapiro, J.L.: The Sensitivity of PBIL to the Learning Rate, and How Detailed Balance Can Remove It. In: Cotta, C., de Jong, K., Poli, R., Rowe, J. (eds.) Foundations of Genetic Algorithms VII. Morgan Kaufmann, San Francisco (2002)
8. Očenášek, J.: Parallel Estimation of Distribution Algorithms, Brno University of Technology, Faculty of Information Technology, Doctoral Thesis (2002)

Adaptive Shot Change Detection for Hardware Application

WonHee Kim, KwangSeok Moon, and JongNam Kim

Dept. of Electronic Computer Telecommunication Engineering, PuKyong Nat'l University, Busan, Korea
{whkim07,moonks,jongnam}@pknu.ac.kr

Abstract. Shot change detection is the main technique for automatic temporal segmentation of video. Until now, there were little research reports about real-time shot change detection for applying to hardware terminals with low performance such as PMPs (Portable Multimedia Player) and cellular phones. In this paper, we propose an adaptive real-time shot change detection algorithm using sub-sampling, weighting variance, and an adaptive threshold. We use weighting variance of frame difference for dividing feature extraction at a changed frame. We set an adaptive threshold using the average of variance from the next one of a last shot change frame to the previous one of a current frame. In the experiments, we obtained the detection rate of about 94.4% in overall accuracy. We verified real-time operation of shot change detection by implementing our algorithm on the TVUS HM-900 PMP from the company of Homecast. Therefore, our proposing algorithm will be helpful in searching video data on portable media player such as PMPs and cellular phones.

Keywords: Adaptive Shot Change Detection, Implementation on PMP, Weighting Variance, Adaptive Threshold.

1 Introduction

Shot change detection is a basic technique and one of important things for management of efficient video data [1]. It is enable to reconstruction, and it has a function to divide and searching for data. Shot change detection is a technique to detect between shots of the video data and boundary [2].

Shot change detection algorithms is approached by a various researched, but now it is not commonly use in real- time processing application. Because of previous algorithms have a defeat which is not robust use in a various genre. Because previous approached is focus on accuracy of detection ratio. In this field, it has a few studies for the reduction of computation. Furthermore the study is not accomplished with device on exception of PC. Like as PMP and most of portable device include Media player function, so it is necessary for the study in real time. Shot change detection algorithms require not only PC but also others devices.

A PMP is needed to real time shot change because of these things. First, PMP has restrictive hardware resource. Second, PMP is supply to restrictive electric power.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 778–784, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Third, the users get an answer immediately. When you use not real-time algorism, you have a complaint because of you have to check all things if users want to next shot change frame. Shot change detection have to real time system that is needed to small cash memory, using the low power consumption and a quick answers.

In this paper, we aim to develop shot change detection algorithm considering real-time application. When real time detection is using the study, light scheme is essential. In order to light scheme adapt the sub-sampling and it has little loss. To meet this condition, we extract a feature of shot change frame using variance of difference frame. Proposed method takes small computation amount and is invariant from sub-sampling. There is no user's command in setting threshold, to process in real-time. To satisfy this term, we set mean of weighting variance in variable block as adaptive threshold.

2 Related Works

Shot change detection algorithms are approached by a various researched in spatial domain and frequency domain. Zhang et al introduced pixel-wise comparison [3]. Pixel-wise comparison is a scheme to use difference intensity for each other pixel at continuous two frames. The block-based scheme focuses on local features that, otherwise, would be sensitive to camera and object activities. Zhang et al introduced likelihood ratio comparison. This scheme decide for shot change's point through a frame by frame, each frame has pair of blocks. The histogram based methods is widely used in detecting shot changes. Tonomura proposed the simplest scheme for detecting threshold-based scene changes through comparing among gray-level histograms. Nagasaka et al proposed a x^2-test scheme for focusing on the difference value between two frames as well as on the motion of a camera or object.

A number of shot change detection methods have been proposed in frequency domain. In order to reduce the high complexity caused by IDCT, the approaches for shot change detection in MPEG compressed domain have been developed. Meng et al use the variance of DC coefficients and motion vectors to analyze shot changes. Zhang et al count the number of valid motion vectors in predictive-coded pictures or P-pictures and bidirectionally predictive-coded pictures or B-picture. Fernando et al use the macro-block prediction statistics of B-pictures. Seong et al simplify the Fernando's method to apply to a hard disk drive embedded digital satellite receiver. Kim et al implemented the shot change detection algorithm using reduced images of only intra pictures in MPEG-2 compressed domain and demonstrated the efficiency of algorithms in a commercial PVR [4].

Another most common method is commonly use edge comparison, normalized inner product based method, image segmentation based scheme [5].

3 Proposed Method

In this paper, we propose adaptive real-time shot change detection technique using weighting variance. In this section, we describe a concept of weighting variance and setting procedural of adaptive threshold.

3.1 Weighting Variance

We used a variance for searching shot change frame. A mean of pixels is calculated from a difference frame of continuous two frames, a variance of next frame is calculated using a mean of pixels. Eq. (1) and Eq. (2) is represented mean of difference frame and variance next frame.

$$m = \frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(f_i(x,y) - f_{i+1}(x,y))}{W \times H} \tag{1}$$

$$v = \frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(f_{i+1}(x,y) - m)^2}{W \times H} \tag{2}$$

m is mean of difference frame and v is variance of next frame. f_i is current frame, f_{i+1} is next frame. W and H is a size of frame. The weighting variance means dividing W and H of denominator and each weighting. Eq. (3) and Eq. (4) is shown this procedure.

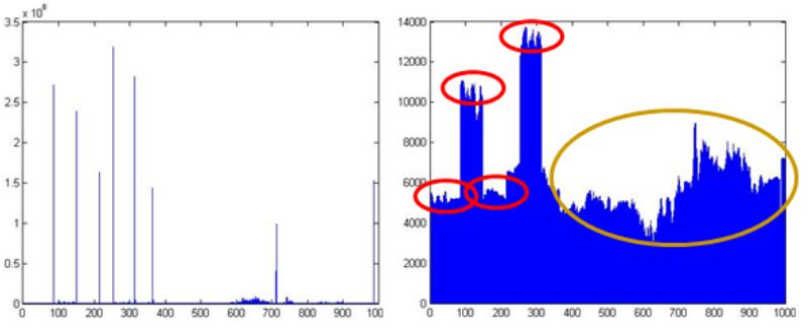
$$m' = \frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(f_i(x,y) - f_{i+1}(x,y))}{(W / w_d) \times (H / w_d)} \tag{3}$$

$$v' = \frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(f_{i+1}(x,y) - m')^2}{(W / w_d) \times (H / w_d)} \tag{4}$$

m' is mean applied weighting, v' is variance applied weighting, w_d is variance weighting. We knew the optimal weighting through the average and dispersion's graph. Seeking the graph dispersion, increasing the w_d 1 to 1. It has a little different graph over the 4. It is effected video sequence, and we checked it is more different when it has a graph over the 4. Compared with shot change frame with no shot change frame has higher dispersion. Also the weight is getting larger, shot change frame is bigger than the others. These values take a kind of exponential function. For the reason the weight is getting larger, the division is clearly and the value is increasing also a system also measuring is more complicated. Moreover, sometimes the value's limits over the data type, it is a critical problem. Namely, it doesn't match the computation amount, but it makes clear the distribution of variance values. And variance of shot change frame must be limited the data type. Purposed methods are considering these things and decide weight.

m' applied weighting is by dividing denominator and weighting. Thus, m' is $w_d^2 \times m$. Also m' and v' are much larger value than v. This value is not linear increasing, is value more than several tens times of non-change frame in shot change frame. Shot change frame is much easier classified by this distribution of value. Fig. 1 described distribution of proposed method and distribution of variance of difference frame. If weighting is applied because much larger value is shown in shot change frame, we know remarkably shot change frame. In contrast in case of general variance not

applied weighting, shot change detection is possible like marked part of small circle, but obscure case happens like marked part of large circle.



(a) Weighting applied (b) Weighting not applied

Fig. 1. Distribution of variance by weighting values

3.2 Adaptive Threshold Setting

Threshold is one of the important factors that it determines detection result. We can know the fact that different threshold is needed in other video data. There is no user's command in setting threshold, to process in real-time. Thus adaptive threshold is required.

Proposed method uses mean of weighting variance values to set threshold adaptively. Eq. (5) is shown this procedure.

$$TH = \frac{\sum_{i=s_p+1}^{s_c-1} v'(i)}{s_c - s_p + 1} \times w_t \tag{5}$$

s_p is previous shot change frame, s_c is current frame, w_t is threshold. Adaptive threshold, TH is defined the value that w_t is multiplies mean of weighting variance of difference frame, from next frame of previous shot change frame to previous frame of current frame. Current frame becomes shot change frame in case variance, $v'(sc)$ in current frame is larger than TH, mean of variance multiplied weighting. To multiply threshold weighting, w_t keeps threshold from setting up very low. Value of w_t is suitable 10~15, variation of value within this section doesn't affect detection result.

4 Experimental Results

The proposed method was tested following conditions. Environment of PC is CPU Pentium-4 2.8GHz and RAM 1G. A detection program is implemented using VC++ .net 7.1 and Matlab 7.3. We used ten video data which have 176×144 size, YUV(4:2:0) format, 9000 frame. Proposed method is evaluated by precision and

recall, F1 [6]. In experimental result of table, the value is presented multiple of 100 in precision and recall.

The proposed method was tested two experiments. Firstly, we compared proposed method(Z) with conventional methods(A~J) at video size of 176×144. The conventional methods have a optimal result using iterative experiments of fixed threshold. Table 1 shows results of first experiment. The proposed method is improved with 2~20.4% in precision, 3~18.2% in recall, 1.1~19.3% in F1, which may vary according to comparison methods. Methods in table 1 have results through correction of threshold of 30 times or more per method to search optimal result. We experiment and obtain result until we search the best result about ten video sequences. In other words, we mean that methods in Table 1 can't apply actually, but can present the best result. So we think that methods in Table 1 are proper to understand level of performance compared proposed methods.

Next, we compared proposed method(Z) with conventional methods(A~I) at sub-sampling video size of 11×9 using the same condition. At second experiments, we excepted algorithms of impossible sub-sampling and methods of bad result value. Table 2 shows results of second experiment. The proposed method is improved with 13.8~57.8% in precision, 12.6~57% in recall, 13.1~57.4% in F1, which may vary according to comparison methods. We confirmed that method of D, E, F which shown a result similar level to proposed method is Table 1 decreased remarkably the result. In sub-sampling video sequence, histogram based methods that generally had good performance existing have lower performance rather than pixel-based methods. We think histogram based methods are not proper if it used in application that it can use only extremely reduced video size.

Table 1. Comparison of performance in 176×144

Methods	Precision	Recall	F1
A : Pixel-wise comparison[3]	84.5	84.5	84.5
B : Likelihood ratio[3]	95.9	95.8	95.9
C : MSE comparison[6]	90.0	90.2	90.1
D : Intensity histogram[3]	93.5	93.2	93.3
E : Color histogram[2]	96.5	96.6	96.5
F : Chi-square test[2]	95.5	95.7	95.6
G : Variance comparison[1]	88.4	88.0	88.2
H : Variance of different histogram[1]	83.4	83.4	83.4
I : Variance of different frame[1]	87.0	87.0	87.0
J : DC-coefficient[4]	75.1	75.0	75.1
K : Cheng's method[7]	86.1	94.3	90.0
L : Ko's method[8]	84.4	99.0	91.1
Z : Proposed Method	95.5	93.2	94.4

On the basis of two experimental results, we could know that proposed method is shown improved result regardless of size of video sequence. Most of methods has much loss in sub-sampling video sequence, but proposed methods has little loss by 2%. Pixel-based methods have loss by about 15%, histogram based methods have loss by about 50%.

Table 2. Comparison of performance in 11×9

Methods	Precision	Recall	F1
A : Pixel-wise comparison[3]	76.0	75.4	75.7
C : MSE comparison[6]	79.1	79.2	79.2
D : Intensity histogram[3]	35.1	34.8	34.9
E : Color histogram[2]	39.6	36.8	38.1
F : Chi-square test[2]	37.4	36.8	37.1
G : Variance comparison[1]	72.0	72.3	72.1
H : Variance of different histogram[1]	43.0	43.2	43.1
I : Variance of different frame[1]	72.7	72.4	72.6
Z : Proposed Method	92.9	91.8	92.3

5 Implementation on PMP

In this paper, proposed method is applied in PMP for verify of real-time processing. It is difficult to test all algorism in PMP, contrast with PC. It is very difficult works amending a code or adding a function when it is finished development. Therefore we gave shape in PC experiment, and we prove proposed methods are fit for PMP. Color-histogram comparison is not proper in real-time processing because has a lot of computation amount. Fixed threshold based methods can not apply in like as PMP because of requiring correction of threshold. Besides original adaptive threshold setting methods are not used in real-time processing because it is needed to whole video sequence information. While proposed method has small computation complexity by 1.1 ~ 1.2 times computation amount of pixel-wise comparison presenting conventional minimum computation complexity. Also, it can use in real application because of automatic threshold setting using adaptive threshold. Moreover, it is different from existing adaptive threshold setting methods, so it doesn't necessary to whole video sequence information, and it is possible to shot change detection if it just needed previous shot change frame information. Therefore proposed method is good at PMP.



Fig. 2. Result of shot change detection on actual PMP

In PMP, a detection program is implemented using WinCE5.0 and Embedded Visual C++ 4.0. When we apply to our method in PMP, it doesn't have false alarm, and we confirmed it doesn't have any factors for the effective in repeating. Furthermore there is no delay and buffering. It just has one frame delay when beginning shot change detection. Therefore we verified our proposed method is implemented in PMP. Detection result is similar to in experiment of PC. Fig. 2 presented result image of shot change detection using the PMP.

6 Conclusions

In this paper, we propose an adaptive real-time shot change detection technique using a weighting variance and an adaptive threshold. We use a weighting variance of difference frame for dividing feature extraction at shot change frame. We defined an adaptive threshold using the average of variance from next frame of previous shot change frame to previous frame of current frame. In the experiments, the result of proposed method is improved compare with conventional methods. At sub-sampling video data for reduce of computation, a loss of proposed method is little. We made certain of the proposed method which is possible to apply TVUS HM-900 PMP of Homecast in real-time shot change detection. Proposed methods will be used helpful in a portable media player like as PMP and a Setup-box.

Acknowledgements

This work was supported from Advanced Technology Project by SMBA and RIS Project by KOTEF.

References

1. Cotsaces, C., Nikolaidis, N., Pitas, I.: Video Shot Detection and Condensed Representation. IEEE Signal Processing Magazine 23, 28–37 (2006)
2. Smoliar, S.W., Zhang, H.J.: Content-Based Video Indexing and Retrieval. IEEE Multimedia 1(2), 62–72 (2006)
3. Zhang, J., Kankamhalli, A., Smoliar, S.W.: Automatic partitioning of full-motion video. ACM Multimedia Systems (1993)
4. Kim, J.R., Suh, S.J., Sull, S.H.: Fast scene change detection for personal video recorder. IEEE Transaction on Consumer Electronics 49, 683–688 (2003)
5. Bescos, J., Cisneros, G., Menendez, J.M., Cabrera, J.: A unified model for techniques on video-shot transition detection. IEEE transaction on Multimedia 7, 293–307 (2005)
6. Boccignone, G., Chinaese, A., Moscato, V., Picariello, A.: Foveated Shot Detection for Video Segmentation. IEEE Transactions on Circuits and Systems for Video Technology 15, 365–377 (2005)
7. Cheng, Y., Yang, X., Xu, D.: A Method for Shot Boundary Detection With Automatic Threshold. Proceedings of IEEE TENCON 1, 582–585 (2002)
8. Ko, K.C., Rhee, Y.W.: Video Segmentation using the Automated Threshold Decision Algorithm. KSCI Journal 10(6), 65–73 (2005)

An Efficient Self Routing Scheme by Using Parent-Child Association for WSNs

Yeon-Mo Yang and IlSoo Jeon

Kumoh National Inst. of Tech. (KNIT), 1 Yangho, Gumi, Gyeongbuk, 730-701, Korea
{yangym,isjeon}@kumoh.ac.kr

Abstract. The IEEE standard 802.15.4 for WPAN (Wireless Personal Ares Networks) shows promise of bringing ubiquitous sensor networking (USN) into reality, at least technically. In this paper, an Association based Self Routing (ASR) scheme of a IEEE 802.15.4 beacon-enabled mode is presented to minimize the control packet overhead and surplus beacon channel power during the routing discovery process in the data transmission for the upstream direction. It also maximizes channel utilization under low duty cycles or traffic congestion. The proposed scheme, ASR routes packets from sensor routers to sink based on the parent-child association relationships established by the IEEE 802.15.4 topology formation procedure. By using the ASR, the network resources for control packets are efficiently utilized and adaptively allocated throughout different duty cycles. Extensive simulation results using TOSSIM (TinyOS mote SIMulator) and experimental results with Mica-Z show that the ASR scheme outperforms well in comparison to the conventional Ad-hoc On-Demand Distance Vector (AODV) routing scheme over the parameters such as: average throughput, number of collisions, number of network commands, and delivery ratio under the upstream data transmission.

Keywords: Wireless personal area networks (WPANs), ad-hoc routing protocol, quality-of-service (QoS), self routing, super frame, beacon-enabled networks.

1 Introduction

According to well known literature, in the next few years, it is highly likely that energy-efficient Ubiquitous Sensor Networks (USNs) or Wireless Sensor Networks (WSNs) will be requested for use in a wide variety of hybrid system applications such as agricultural or environmental monitoring; caring for the elderly people with emergency and disaster response; 3D (three-dimensional) entertainment and network-based toys; industrial facility sensing and actuator control; Location Based Service (LBS) including logistics and asset tracking; and building automation [1-14].

In these applications, many embedded devices operating on small limited-power batteries are deployed in an area communicating through wireless radio channels. Compared to existing Wireless Local area Networks (WLAN) which focus on providing relatively high-throughput, low-latency for fast file transfer, video conference, and

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 785–794, 2008.
© Springer-Verlag Berlin Heidelberg 2008

real time multimedia applications, the required data rate for body area network (BAN) or wireless personal area network (WPAN) applications is expected to be only on the order of tens of kilo bits per seconds (kbps). Similarly, the required message latency may be on the order of 100ms or more [5-7]. The typical example of WPAN topology is shown in Fig. 1. However, the key concern in these applications is that of the extremely low power consumption, since it is often infeasible or undesirable to replace or recharge batteries for the devices on a regular basis due to their remote locations and cost of deploying.

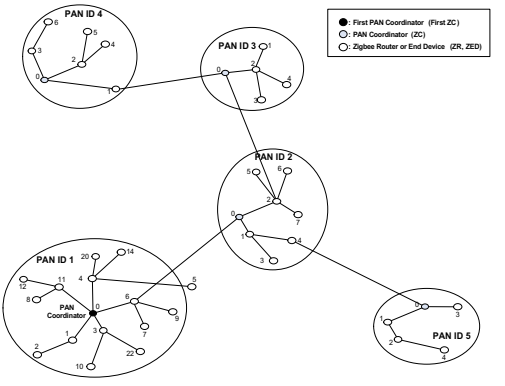


Fig. 1. The system architecture of a WPAN

With respect to previous WPAN standards such as Bluetooth [1], the IEEE 802.15.4 (referred to as 802.14.4 hereinafter) standard is characterized by ultimately lower power consumption and relative lower cost. While, in line with the family of IEEE 802 standards, the standard focuses mainly on PHYsical (PHY) and Medium Access Control (MAC) layers, upper layers of the protocol such as routing (layer 3: L3) or transportation (layer 4: L4) layers are defined by the ZigBee Alliance [15-17], which also deals with the interoperability of WPAN devices from different manufacturers. The MAC protocol plays a significant role in determining the efficiency of the wireless channel bandwidth sharing and the energy cost communication. The relationship between 802.15.4 and ZigBee is similar to that between IEEE 802.11 and the Wi-Fi alliance. Based on 802.15.4, ZigBee has proposed a specification for a suite of high-layer communication protocols [15-18].

802.15.4 can operate at data rates of 250 kbit/s (less than 250 kbit/s) at 2.4GHz, 40 kbit/s at 915MHz (North America), and 20 kbit/s at 868 Mhz (Europe). While the data rate is much lower than Bluetooth, the energy consumption is at least one order of magnitude lower and recent low-cost commercial versions have demonstrated the viability of this technology for low duty cycle (less than 1%) sensor applications. Significantly, the latest generations of motes from UC Berkeley with Crossbow, i.e., Mica-Z is based on the primitive 802.15.4 compliant TI Chipcon CC2420 radio [2], [3].

In this paper, motivated by the above trends, we consider an 802.15.4-based WSN and we study the relationship between the 802.15.4 topology formation mechanism

and possible routing strategies at the network layer. Our objective is to explorer the cross-layer interdependencies between the topology formation like association information and routing mechanisms to set-up energy efficient routing paths towards a sink.

To this aim, we have proposed a simple Association-based Self Routing (ASR) for the upper stack of 802.15.4 sensor networks. To verify and validate the behavior of the proposed scheme, we have compared and contrasted the performance of ASR to Ad-hoc On demand Distance Vector (AODV) [14] scheme which was designed for highly dynamic or mobile application scenarios. ASR is a kind of simple parent-child routing scheme where ASR routes packets in a upstream direction from sensors to sink, a proactive routing, which is based on the parent-child relationships established by the 802.15.4 MAC topology formation procedure. Basically ASR and AODV are inherently different, as the former is a proactive routing scheme that does not use ad hoc control messages in upstream direction, while the latter is a pure on-demand route acquisition algorithm, which broadcasts discovery packets when a routing path needs to be established.

To the best of our understanding, except [10] which only considers IEEE 802.15.4 association tree in an upstream direction, none of the previous studies have addressed utilizing the parent-child relations and the BOP [16] for the routing path discovery in a WPAN under TinyOS, which triggered the motivation for this work. First, the ASR confines the number of retrial in the joining process in the aims of reducing unwanted noise which occurs during the association phase. Basically, the association process causes a beacon request message to find channels in active scan mode, it consumes a lot of resources. Thus in ASR it could be possible to limit beacon request flooding as much as possible; second, the ASR exploits information exchanged during the network formation and topology update phases, thus avoiding additional routing messages and the associated overhead.

2 Protocol Descriptions: Association Based Self Routing

The proposed scheme, ASR, is explained in the directions of the upstream, the data transmission from each end node to the sink node, ZR0 (ZigBee Router 0). First, ASR has setup the routing path from the source node to the destination node without sending a flooding message, route request (RREQ). In a previous scheme, AODV, it extensively relied on sending an RREQ message to find the routing path or flooding to setup the path, causing consumption of a lot of bandwidth resources. Compared with this, we have proposed ASR to suppress the surplus flooding during the routing path discovery process.

Based on the ZigBee specifications, we have proposed an Association based Self Routing (ASR) that utilizes the MAC layer associations to perform the routing function, the destination routing path. Under the ASR scheme, the data generated by the end nodes are forwarded to the sink node. Then, the data are forwarded upward to the root tree by means of the parent child relationships. The proposed ASR is a table based routing because the destination path is resolved before an end node performs frame forwarding. The established routing path is the association tree that is formed during the MAC association procedure. As a result, routing paths are relying on

MAC layer parent child relationship and result in a routing path to the ZR0, e.g., the sink node in a PAN [10]. Based on this observation, to collect data generated by the end nodes at the sink node under a multi-hop network, each end node or sensor nodes shall transmit all the packets to ZR0. For ASR, some notations have been defined as following:

- ZR0: the ZigBee coordinator or sink node.
- ZRn: the nth ZR or end node where n id any none zero number less than 255.
- Level or depth of each node: the number of hops from ZR0 to each node, ZRn.

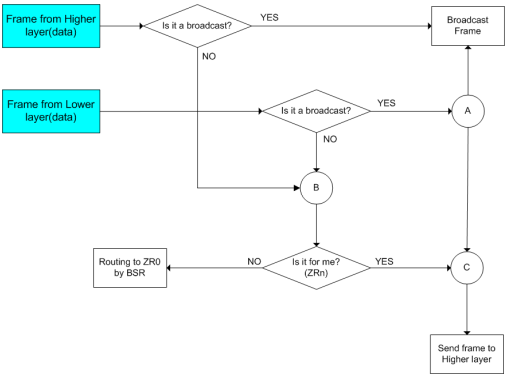


Fig. 2. Association based Routing

Based on the above notation, every node routes all packets directed to the sink ZR0 toward its parent. We remark that the resulting routing paths do not change while synchronization between device and coordinator is maintained, and will be eventually updated after orphan scan calls, e.g., in case of loss of synchronization.

Table 1. Association based routing table

Parent Address (10)		Child Address (10)	
Short, 16bit	Long, 64bit	Short, 16bit	Long, 64bit
0x01	0x0000000000000001	0x09	0x0000000000000009
		0x0a	0x000000000000000a
0xfe	0x0000000000000004	0x04	0x0000000000000004
0xff	-	-	-

The ASR routing algorithm drastically reduces the number of routing control message request since it utilizes and gets the routing information from MAC association procedures. A cross-layer consideration is thus systematically reflected

between routing layer and MAC layer. Thereby, ASR is relatively simple, light and only consumes a small system memory for maintaining the routing tables. Specifically, these features make ASR a good candidate for a small sensor node like MICA-Z.

During data communication when there is a broken in routing path, the automatic routing path recovery process is invoked by means of ad hoc path recovery. The scheme is based on continuous monitoring the number of routing failure count. When the number of routing failure has reached the predefined value, it automatically reinitiates the association process as explained the above. Fig. 2 shows the association based routing when a news packet has come and Table 1 shows an example of association based routing information.

3 Performance Evaluation

We now describe our performance evaluation study and experimental results of the 802.15.4 MAC. To assess the performance of the proposed parent-child routing scheme, we implemented ASR and AODV in the TOSSIM network simulator related to Mica-Z with the SKT standard wireless extension [16]. Tiny-OS provides non-commercial the 802.15.4 physical and MAC layers from public domain and of several routing protocols which comes from CUNY [7]. Based on this, we implement a cross-layer interaction between link (layer 2: L2) and routing (L3) layers to route data packets to the coordinator (ZR0) with which each device has been successfully associated. The address of ZR0 is available at MAC layer in the association response command frame sent by ZR0 to its child and it is forwarded to the network layer which inserts it in the routing table as next-hop for packets destined to the sink. The network topology used in the simulation is a 6 hops binary tree (i.e. 64 nodes) with 20m distance between adjacent nodes. The node at the center is the ZR0. Each other node is in the radio transmission range of the others and transmission to receiving radio characteristic is symmetric for simplicity.

3.1 Model Description and Performance Metrics

We consider a network consisting of N = 64 sensor nodes distributed on a 160 x 160m grid. All sensors are routers or FFD (ZRn except ZR0) devices configured in beacon-enabled mode, and they are all potential coordinators in the WPAN allowing for the association of other devices. We consider a single sink (ZR0) located around the center of the area (x = 80m and y = 80m). The network topology is a binary tree, under the given architecture; all routers could generate data in a period manner that will be gathered at the sink in upstream direction. The maximum of hops from ZR0 to the end node is 6 where the total number of nodes are 2^6-1 = 63. During data transmission, any routers could participate in the multi-hop relaying process, if necessary. The traffic generated by the source nodes depends on scenarios such as increasing hops and changing duty cycles. The whole frame size is as follows: MAC header is for 8 bytes, network header is for 11 bytes, frame checker for 2 bytes, and user payload is 36

bytes, so in total is 57 bytes. Packet frame frequency rages from 10 pps to 200 pps per node during experiments. Thus, the average generation data rate equals to around 4k ~ 20kbit/sec (bps).

We remark that once the ZR0 starts the WPAN, network formation, assigning beacon interval and duty cycles, and initiating MAC joining process between ZigBee routes. The performance analysis presented in the next section has been carried out in 3 scenarios such as increasing hops and changing duty cycles. First, increasing the number of hops has gained by deploying more nodes in a grid. Second, changing duty cycles was done by changing BO and SO values as follows: 12.5% to 8:5, 25% to 8:6, 50% to 8:7, and 100% 8:8.

We measure different performance metrics. In particular, for each source we evaluate, the performance metrics are the following:

1. Throughput (kbps): it means the amount of data received by the destination node within certain period of time. In multi-hop environment, the throughput is computed at the final destination as the intermediate nodes are responsible of relaying the packets.
2. Packet delivery ratio (%): it measures the ratio of total packets received against total packets sent in the MAC sublayer. In other words, the ratio of packets successfully received in packets sent in MAC sublayer. This metric does not differentiate transmissions and retransmissions, and therefore does not reflect what percentage of upper layer payload is successfully delivered, although they are related.
3. Collision rate: the total collision during a simulation run. Specifically, this means the total collisions that occur between hidden terminals during a simulation run. Hidden terminals prevent carrier sense from working effectively, and therefore transmissions from them are likely to collide at a third node [15]. In 802.11, the request-to-send (RTS) and clear-to-send (CTS) mechanism is used to tackle this problem [18].
4. Number of network commands (RREQ flooding): number of route request control messages generated by sources to discover paths towards the sink (only for AODV routing).

3.2 Experimental Results and Discussions

We have compared the proposed scheme, ASR and the well known AODV. We carried out an extensive experimental analysis to compare ASR and AODV. The proactive scheme, ASR, compared with a reactive mechanism, AODV, shows better performance with respect to several metrics such as average throughput, number of collisions, number of network commands, and delivery ratio. Therefore, it is able to simply set-up routing paths towards a sink (ZR0) by exploiting the cross-layer interdependencies between the topology formation and the routing mechanisms. ASR exploits information exchanged during the network formation and topology update phases, thus avoiding additional routing messages and the associated overhead. Moreover, it shows several performance enhancements with respect to AODV, such as reduced latency and number of network commands. Finally, it reduces complexity, as it is very easy to implement and does not require a specialized daemon on the host device where it runs. In general, AODV shows in general worse performance.

However, since it does not strictly depend on the MAC layer procedures, it provides greater flexibility and can be easily extended to account for different route selection metrics. In general, AODV may be selected when the application layer requires high flexibility from the routing mechanism, on the other hand ASR, thanks to its simplicity, is a light routing protocol for simple application scenarios.

Figures 3(a) and (b) illustrate the comparison of average throughput and delivery ratios with respect to (w.r.t.) ASR and AODV. Figure 4 (a) and (b) illustrate the comparison of number of networks commands and number of collisions w.r.t. ASR and AODV. The figures show the relationship between the number of hops (independent parameters); and the average throughput, delivery ratio, number of network packets, and number of collision (dependent parameters), respectively. Increasing the number of hops was gained by deploying more nodes in a grid as the network scope expended.

When the hop count is a little smaller, both throughput and delivery ratio are less different, but in a larger case than 4, it is much different. Basically, for both the schemes, the delivery ratio goes down to around 50% with an increasing hop counts due to the increasing number of collisions on the channel and higher collision probability. However, when the hop count is larger than 6, with ASR the packet loss is very low (from 20% to 25%) but with respect to AODV (up to 50% or 60%). In fact, as previously discussed, ASR exploits the 802.15.4 association procedure for routing; and also it does not use RREQ control messages and suppress beacon request mechanisms during passive scan process, which are the main causes of collisions and cause surplus power consumption. Moreover, from the comparison of Fig. 3 and Fig. 4, it can be noted that the performance of BSA basically does not significantly change for different hop counts; wile in the AODV case, packet loss varies for changing hop counts. This is due to data packets collisions that occur during the whole duration of the event, and cause routing paths to fail and retry finding the route discovery mechanism.

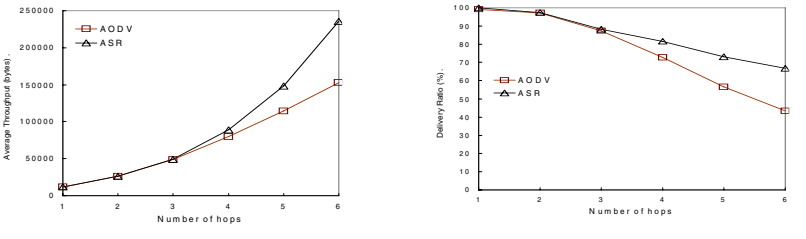
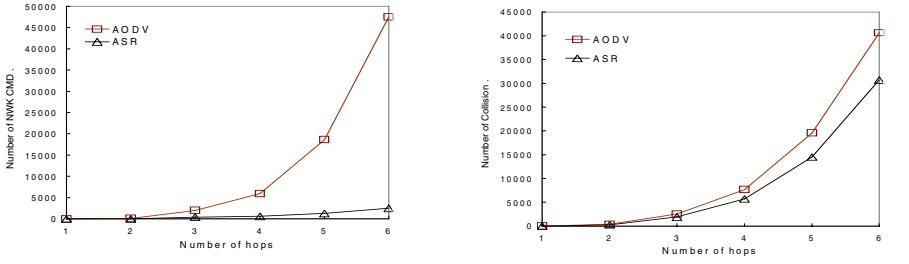


Fig. 3. Comparison between ASR and AODV for, a) average throughput and b) average delivery ratio with different number of hops

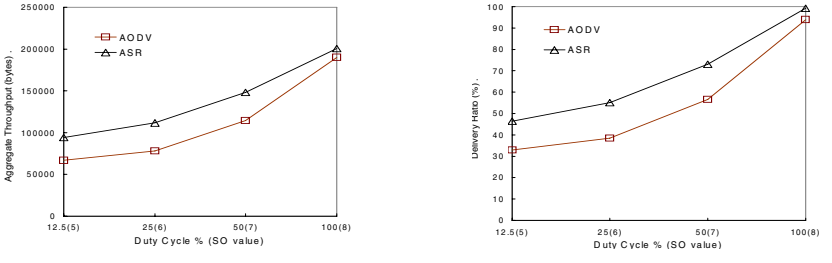
To better point out the congestion caused by the route discovery process, in Fig. 4 we report the total number of network commands as a function of for different values of hop counts. In the worst case, i.e., when the hop count is larger than 6, AODV produces about 45000 network commands, while ASR is less than 200. Moreover, the high load of routing control messages also causes route discovery failures, which result in the steep slope of the curves.



(a) (b)

Fig. 4. Comparison between ASR and AODV for, a) average number of network commands and b) average number of collisions with different number of hops

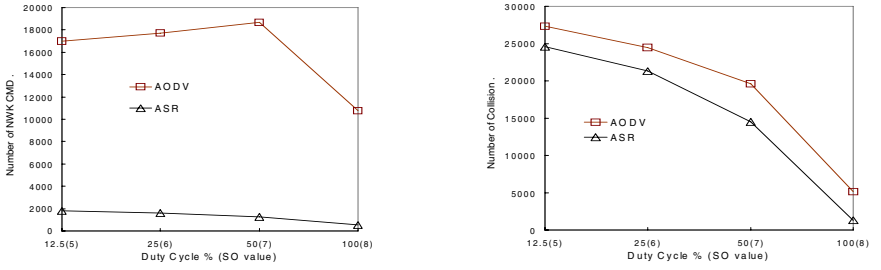
Figures 3(a) and (b) show the comparison of average throughput and delivery ratios w.r.t. ASR and AODV. Figure 5 (a) and (b) show the comparison of number of networks commands and number of collisions w.r.t. ASR and AODV. The figures show the results of performance metrics when the duty cycles have changed. Duty cycles have changed to the value of BO and SO as by changing SO from 5 to 8. When the duty cycle is relative low, the throughput of ASR is much larger than AODV. This is well explained by Fig. 6(a) that indicates that reducing RREQ flooding reduces packet collision through multiple hops.



(a) (b)

Fig. 5. Comparison between ASR and AODV for a) average throughput and b) delivery ratio with different duty cycles

When there is congestion because of a low duty cycle such as 12.5 or 25%, in ASR the metrics show 20% less than that of AODV. As shown in Fig. 5(b), and Fig 6(b), specifically, in an upstream direction, association based routing affects to channel utilization, reducing the overheads significantly in ASR. The performance metrics are about 10% less than that of AODV scheme through out changing duty cycles.



(a) (b)

Fig. 6. Comparison between ASR and AODV for a) of number of network commands and b) number of collision with different duty cycles

4 Conclusions

We have proposed a new 802.15.4 Association-based Self Routing (ASR) scheme. Within ASR it considers utilizing the parent-child relationship for establishing a routing table in an upstream direction, and suppressing surplus beacon power during the channel association process. Practically, ASR shows a dramatic increase in network performance when traffic congestion level is severe or the duty cycle is relatively low. Furthermore, it minimizes control packet overhead during routing discovery and beach channel power in channel scanning processes. The experimental results with TOSSIM with Mica-Z and evaluation facts coincide to each other and make sure the proposed ASR outperform the previous ad-hoc routing algorithm like AODV in terms of stabilization time, control packet overhead, and channel utilization.

Acknowledgments. The author would like to thank Dr. Dongkyun Kim of KNU, Dr. Young-Jin Nam, Young-Duk Kim, and Dr. Dong-Ha Lee of DGIST for their fruitful discussions and valuable comments. This paper was supported by Research Fund, Kumoh National Institute of Technology (www.kumoh.ac.kr).

References

1. Liu, Y., Lee, M., Saadawi, T.: A Bluetooth Scatternet-Route Structure for Multi-hop Ad Hoc Networks. IEEE Journal on Selected Areas in Communications 21, 229–239 (2003)
2. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In: SenSys 2003 (2003)
3. J. Polastre, J. Hill, and D. Culler: Versatile Low Power Media Access for Wireless Sensor Networks. In: SenSys 2004 (2004)
4. Lu, G., Krishnamachari, B., Raghavendra, C.S.: Performance Evaluation of the IEEE 802.15.4 MAC for Low-rate Low-Power Wireless Networks. In: IEEE Int. Performance Computing and Communication (2004)
5. Zheng, J., Lee, M.L.: A Comprehensive Performance Study of IEEE 802.15.4. IEEE Press Book (2004)

6. Koubaa, A., Alves, M., Tovar, E.: A Comprehensive Simulation Study of Slotted CSMA-CA for IEEE 802.15.4 Wireless Sensor Networks. In: IEEE International Workshop on Factory Communication Systems (2006)
7. Lee, M., Zheng, J., Hu, X., Juan, H., Zhu, C., Liu, Y., Yoon, J., Saadawi, T.: A New Taxonomy of Routing Algorithms for Wireless Mobile Ad Hoc Networks: The Component Approach. IEEE Communications Magazine, 116–123 (2006)
8. Koubaa, A., Alves, M., Attia, M., Nieuwenhuyse, A.: Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks. Technical Report IPP-Hurray, TR-061104 (2006)
9. Suh, C., Ko, Y., Lee, C., Kim, H.: Numerical Analysis of the Idle Listening Problem in IEEE 802.15.4 Beacon-Enable Mode. In: Proc. of ACM ChinaCom 2006 (2006)
10. Cuomo, F., Della, L., Monaco, U., Melodia, F.: Routing in ZigBee: Benefits from Exploiting the IEEE 802.15.4 Association Tree. In: ICC 2007, pp. 3271–3276 (2007)
11. Mura, M.: Ultra-Low-Power Optimizations for the IEEE 802.15.4 Networking Protocol. In: IEEE MASS 2007, pp. 1–9 (2007)
12. Chen, F., Wang, N., German, R., Dressler, F.: Performance Evaluation of IEEE 802.15.4 LR-WPAN for Industrial Applications. IEEE WONS, pp. 89 – 96 (2008)
13. Pollin, S., Tan, I., Hodge, B., Chun, C., Bahai, A.: Harmful Coexistence Between 802.15.4 and 802.11. In: IEEE CrownCom, pp. 1–6 (2008)
14. Royer, E.M., Perkins, C.E., Das, S.R.: Ad hoc On-Demand Distance Vector (AODV) Routing (RFC 3561)
15. IEEE P802.15.4: Wireless Medium Access Control and Physical Layer specification for Rate Wireless Personal Area Networks. IEEE(September 2006)
16. SKT WiBEEM Specification Document (May 2007), `http://www.sktelecom.com`
17. ZigBee Specification Document 053474r17 (October 2007)
18. IEEE P802.11, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE (June 2007)

Cryptanalysis of Transposition Cipher Using Simulated Annealing Genetic Algorithm

Jun Song[1,2], Fan Yang[1,2], Maocai Wang[1], and Huanguo Zhang[2]

[1] Computer School, China University of Geosciences, Wuhan 430074, China
[2] Computer School, Wuhan University, Wuhan 430079, China
songjun@cug.edu.cn

Abstract. The cryptanalysis technology adopting intelligent computing is the combination of intelligent computing and cryptography. Cryptanalysis of the transposition cipher with evolutionary algorithm has attracted much interest in recent years. This paper presents the automatic analysis and exploration on a typical transposition cipher, based on improved fitness evaluation weight mode and using SAGA method. Compared with the earlier experimental results of the GA/TS/SA methods adopted, the attack against transposition ciphers using improved SAGA can achieve comparatively better results on the amount of key recovery.

Keywords: cryptanalysis, intelligent computing, fitness evaluation weight mode, SAGA.

1 Introduction

Cryptology is one of the most enabling technologies to support information security, and the automatization of cryptography and cryptanalysis by employing intelligence optimization technology is a very important and interesting research subject in cryptology study. The cryptanalysis technology adopting intelligent computing is the combination of intelligent computing and cryptography, which is the developing tendency for the automatization of cryptanalysis.

In past years, many researchers in the field of Cryptology and Intelligent Computing are interested in developing automated attacks on transposition cipher. In 1993, Robert Matthews showed that simple transposition cipher could be attacked using a customized genetic algorithm, also called GENALYST [3]. In 1994, Andrew Clark, based on heuristic search method that contains GA (genetic algorithm), TS (tabu search algorithm) and SA (simulated annealing algorithm), carried out one of the most significant studies on transposition cipher [4]. In 2003, the same methods were used by Dimovski etc. [5]. In 2003, another paper showed that simple transposition cipher could be attacked using ant colony algorithm by Russell [6]. In 2005, the scatter search technology was applied to simple transposition cipher by Garici etc. [7]. In 2007, Toemeh etc. well studied how the genetic algorithm was applied to the Cryptanalysis of simple transposition cipher [8].

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 795–802, 2008.
© Springer-Verlag Berlin Heidelberg 2008

John A Clark's invited paper [2] showed the comprehensive reviews for cryptanalysis of evolutionary techniques in IEEE CEC2003. All the past work described that most major optimization techniques could have been applied to attack classical ciphers successfully [2]. It is feasible for classical ciphers to utilize their weak capabilities and low complexities, while for modern block ciphers such as DES, it is more difficult to break because of designing of high nonlinearity and low autocorrelation. In general, most classical cryptosystems are based on linearity mappings such as substitution. It is easy to find a direct or indirect correlation between plaintext and ciphertext in the process of evolution analysis for classical ciphers.

In this paper, we make the automatic analysis and exploration on typical transposition cipher, design and experiment on our improved fitness evaluation weight mode and using SAGA method. In 2003, Dimovski showed the intelligent analysis of the transposition ciphers by the GA/TS/SA methods [5]. Compared with the experimental results that Dimovski gained, the attack against transposition ciphers using improved SAGA can achieve comparatively better results on the amount of key recovery.

The remainder of this paper is organized as follows. In section 2, we give the description of transposition cipher, the notion of the cryptanalysis by GA/TS/SA methods Dimovski adopted [5] and the fitness evaluation weight mode of Toemeh [8]. In section 3, we present our improved fitness evaluation weight mode and strategy of cryptanalysis using SAGA. In section 4, the cryptanalysis experimental results of transposition cipher using SAGA are reported. In section 5, conclusions are derived and directions for future work are suggested.

2 Preliminaries

In order to describe the problem with cryptanalysis of transposition cipher using SAGA, we briefly introduce the structure of transposition cipher, the fitness function of the GA/TS/SA methods Dimovski [5] adopted and the fitness evaluation weight mode of Toemeh [8].

2.1 Transposition Cipher

Transposition ciphers, also called permutation ciphers, are an important family of classical cipher. In transposition cryptosystem, it is changed the mode of iterative letters and the capability of multi-letters combinations in encrypted ciphertext, but the statistics frequency of letters between plaintext and ciphertext is unchanged. That is, a transposition cipher transforms a plaintext by rearranging the positions of the letters of the plaintext without changing the identities of the letters [10]. It is an important clue for designing the fitness function of automatic analysis on transposition cipher.

Let L be a fixed positive integer representing the size of a plaintext/ciphertext block and let K be all permutations of $(1, 2, \ldots, L)$. Since the search space of the exhaustive attack on transposition cipher is $L!$ Where P represents a plaintext block, C represents the corresponding ciphertext block, $P= x_1, x_2,\ldots, x_L$ and $C= y_1, y_2,\ldots, y_L$. Then a permutation $\pi= (\pi(1), \pi(2),\ldots, \pi(L))$ is a key $\pi \in K$, the encryption algorithm of transposition cipher is [1,10]

$$C = E_\pi(P) = x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(L)}.$$

Let π^{-1} denote the inverse of π. The corresponding decryption algorithm of this transposition cipher is

$$P = D_\pi(C) = y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \ldots, y_{\pi^{-1}(L)}.$$

Because the cipher doesn't change any of the letters, the ciphertext will have exactly the same letter frequencies distribution as the underlying plaintext in natural-language-based information. In general, transposition ciphers are stronger if the key is long and follows certain conditions. However, since the identities of the letters do not change, as mentioned above, transposition cipher is also vulnerable to the frequency analysis techniques [1,10].

2.2 The Fitness Evaluation Weight Mode

It is well known that letters in a natural language have stable frequencies. The knowledge of the frequency distribution of letters in a natural language provides significant clue for cryptanalysis of transposition ciphers. Among the methods of intelligent analysis on transposition ciphers, the frenquencies of two-letter combinations, called bigrams, and three-letter combinations, trigrams, usually have been used. The fitness evaluation weight mode is an intuitional and effective method for intelligent cryptanalysis of transposition ciphers.

In 1993, Matthews described a fitness evaluation weight mode for genetic algorithm [3]. Clark improved the Matthews's mode further in next year [4]. In 2007, Toemeh refined on previous work, mainly based on Matthews's mode and Clark's mode, to make it more sufficient and more exact [8]. The fitness evaluation weight scores of Toemeh's mode [8] is illustrate as follow.

Multi-grams	*Score*	*Multi-grams*	*Score*	*Multi-grams*	*Score*	*Multi-grams*	*Score*
E_	+2	_A	+1	EEE	-5	HE_	+5
_T	+1	S_	+1	ING	+5	AND	+5
HE	+1	__	-6	_TH	+5	___	-10
TH	+1			THE	+5		

2.3 The Cryptanalysis by GA/TS/SA and Results

In 2003, Dimovski represented the intelligent ciphertext-only analysis of the transposition ciphers by GA, TS and SA methods [5]. It is typical for the fitness function designed by Dimovski in previous papers, which is same as clark's [4]. The fitness function is defined by [5]

$$C_K = \beta \cdot \sum_{i,j=\Lambda} \left| K^b_{(i,j)} - D^b_{(i,j)} \right| + \gamma \cdot \sum_{i,j,k=\Lambda} \left| K^t_{(i,j,k)} - D^t_{(i,j,k)} \right| + \lambda \cdot \sum_{i,j,k,g=\Lambda} \left| K^f_{(i,j,k,g)} - D^f_{(i,j,k,} \right| \quad (1)$$

Where Λ is set of the elements of letters in English alphabet, k is the key, b, t and f are the statistics of bigrams, trigrams and four-grams respectively. K denotes the frequency of known letters, and D denotes the frequency of the results that ciphertext is decrypted by tested key. The value of β, γ and λ is corresponding n-gram evaluation weight assigned.

The experimental results [5] of the GA, SA and TS methods Dimovski adopted were described by table below. We define the amount of key recovery M and the size of key L. The average results are illustrated by the table when M is 200, 400, 600, 800, 1000 and L is 15, and when M is 1000 and L is 15, 20, 25, 30.

L=15				M=1000			
M	SA	GA	TS	L	SA	GA	TS
200	5	7	4.75	15	12.75	13.25	13
400	10.75	11.5	9.25	20	16.5	17	16.75
600	11.25	12.5	11.5	25	20.15	21.5	21
800	12.5	13	12.25	30	25	25.25	25.5
1000	12.75	13.25	13				

3 Cryptanalysis of Transposition Cipher Using SAGA Method

In this paper, we adopt SAGA (simulated annealing genetic algorithm) for cryptanalysis of a sort of transposition cipher based on ciphertext-only attack. Hereinto, the simulated annealing operation is regarded as an operator. The all-round search capability is better for GA, but the local search capability is bad and vulnerable to a prematurity result. It's just the opposite, for SA. However, SAGA integrates the advantage of both GA and SA into search procedure [11].ï

3.1 Evolutionary Strategy and Parameters

A cryptanalysis algorithm based on intelligent technology is generally composed of followed parts.

1. *Fitness function*. It is, however, a primary problem how to define fitness function. In general, it is feasible for classical ciphers that the design of the fitness measure is modeled on the linear approximate expression or the statistics frequency properties of letters. In this paper, the fitness function of SAGA algorithm is same to Dimovski's.

2. Initialization population. Generally, cryptanalysis using genetic algorithms is stochastic adaptive processes that start with a population of randomly generated candidates and evolution towards better solutions. Hence, the initialization process must comprise every different individual, such as the best, the average and the weak. Therefore, the magnitude of population is large enough for the diversity of chromosomes. It is recommended that the size is over 100.

3. Key representation. The cryptanalysis using intelligent algorithm is kernel to find the optimum key that is similar or equivalent with the true key. As what we do in the character genetic algorithm, the key representation is a chromosome as well, which is also a character pattern. Hence the evolution process of a chromosome is the same as the process of a key optimization.

4. *Selection, crossover, and mutation process*e. While initializing a population of chromosomes, each one with a fitness value, the genetic algorithm will progress by randomly selecting two for mating. The chromosome's selection, crossover, and mutation processes become the same as the processes of optimizing key. Therefore, the selection is weighted in favor of chromosomes with a high fitness value. After the new generation has been determined, the chromosomes are subject to a low rate mutation function which involves different processes in an attempt to vary the genes. The mutation process is inverting the order of a set of letters between two random points. They can help to prevent the algorithm from being stuck in a local optimum point. In general, the mating rate value is better between 0.70 and 0.80, the mutation rate is below 0.001.

3.2 The Improved Fitness Evaluation Weight Mode

The designing of fitness evaluation weight mode is an important factor to attack transposition cipher. It is still comparatively limited for the scope of bigrams and trigrams in Toemeh's mode [8], hence difficult to express the characteristic of various words and phrases. In order to speed analysis process up, we add trigrams and four-grams combinations in improved evaluation weight mode which comprises of Toemeh's mode [8] and the mode of table 1. It is illustrated as below table 1 that trigrams and four-grams combinations are in the evaluation of weight mode.

Table 1. The improved evaluation weight mode (trigrams and four-grams)

Multi-grams	*Score*	*Multi-grams*	*Score*	*Multi-grams*	*Score*	*Multi-grams*	*Score*
THE	+5	ENT	+5	FOR	+5	HIS	+5
ING	+5	THA	+5	DTH	+5	WERE	+8
AND	+5	NTH	+5	HAT	+5	WITH	+8
HER	+5	WAS	+5	SHE	+5	THIS	+8
ERE	+5	ETH	+5	ION	+5		

3.3 The SAGA Algorithm

The SAGA algorithm for cryptanalysis of transposition cipher is as follow:

1. The initialization process: where L is length of key, M is the size of initialization population, G is the current generation, and G_{max} denotes maximal generations. In our experiment, the initialization population P_{old} is randomly generated, and a fitness value for each chromosome in the population is computed according as (1). The T_0 is defined as original temperature, and the factor ΔT for decreasing temperature. Let initial $T = T_0$.

2. The following steps will be repeated until a fixed number of G_{max} generations.

1) New parent chromosome selects $M/2$ pairs test keys from P_{old}.

2) The new population P_{new} is produced from crossover operation. Then selected M chromosomes mutate and compute the fitness value for each test key in the population P_{new}.

3) The simulated annealing operation, adopting a simple decreasing temperature method $T=G\times\Delta T$, is applied to a new population P'_{new}. Then selected M chromosomes mutate and compute corresponding fitness value of each test key in P'_{new}.

4) The M better keys are chosen from P_{old}, P_{new} and P'_{new} according to the successive fitness value and produce current population P_{old}.

3. Fitness function (1) and table 1 are adopted to measure decryption results by trigrams and four-grams fitness evaluation weight mode. Updating and sequencing the list of M better keys across the population P_{old}, the best chromosome is regarded as the true key.

4 Experimental Setup and Results

We experiment with the above SAGA algorithm and improved evaluation weight mode. Based on the above experimental results [5] of the GA and SA methods Dimovski gained, the average amount of key recovery both Dimovski's method and SAGA method adopted is described by table 2. Where M denotes the amount of known ciphertexts, and L denotes the size of key.

Table 2. The average amount of key recovery

L=15				M=1000			
M	SA	GA	SAGA	L	SA	GA	SAGA
200	5	7	6	15	12.75	13.25	14.5
400	10.75	11.5	11.25	20	16.5	17	17.5
600	11.25	12.5	11.75	25	20.15	21.5	22.5
800	12.5	13	13.5	30	25	25.25	27.5
1000	12.75	13.25	14.5				

Table 2 shows the experimental results under the condition that L=15 and M=1000. Compared with the analysis results of the GA and SA methods Dimovski adopted [5], the attack against transposition ciphers using improved SAGA can achieve comparatively better results in the amount of key recovery under the condition that when L is 15 and M=200, 400, 600, when M is 800 and 1000, and when M is 1000 and L =15, 20, 25, 30.

5 Conclusion and Future Work

In this paper, we adopt ciphertext-only attack and for cryptanalysis of a sort of transposition cipher, and add the evaluation weight of trigrams and four-grams combinations in the fitness evaluation weight mode of Toemeh [8]. In 2003, Dimovski adopted the intelligent analysis of the transposition ciphers by the GA/TS/SA methods [5], the fitness function design of which used the statistical frequency of bigrams, trigrams and four-grams. Compared with the experimental results of the GA/TS/SA methods Dimovski adopted, the attack against transposition ciphers using improved SAGA can achieve comparatively better results on the amount of key recovery.

As we discuss above, the intelligent cryptanalysis research on transposition ciphers is comparatively extensive and popular, but there still exists some general problems for most of analysis methods as followed. Firstly, the sampling of real plaintext or ciphertext is much diverse, the frequency of multi-grams such as bigrams or trigrams in ciphertext concealed and homogenized, and therefore high dependency of ciphertext sampling. Secondly, there are still few semantic errors in plaintext that ciphertext is decrypted by selected randomly a better result key. It is need to analyze and determine the best key artificially. Thirdly, the fitness evaluation weight mode is from experimental results, lack of theoretical analysis. In general, it is hard to design a fitness evaluation weight mode that reflects the frequency distribution of letters in natural language well for various plaintext or ciphertext sampling.

The above results indicate that this SAGA algorithm to cryptanalyze transposition cipher is successful and efficient. Contrast with the case of Dimovski's experimental results [5] reported earlier, this research result shows that SAGA algorithm is also a powerful tool for breaking the transposition cipher. Furthermore, we will utilize SAGA algorithm to deal with more complicated transposition cipher for further work in our future.

Acknowledgments. The authors wish to thank the support by Excellent Young Teachers Program of China University of Geosciences (Grant No. CUGQNL0820), Chinese Natural Science Foundation (Grant No. 60373087, 60473023 and 60673071), and 863 Project (Grant No. 2006AA012442 and 2007AA012411).

References

1. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1999)
2. Clark, J.A.: Nature-Inspired Cryptography: Past, Present and Future. In: CEC 2004 (2004); Special Issue
3. Matthews, R.: The Use of a Genetic Algorithm in Cryptanalysis. Cryptologia XVII(2), 187–201 (1993)
4. Clark, A.: Modern Optimisation Algorithms for Cryptanalysis, pp. 258–262. IEEE, Los Alamitos (1994)
5. Dimovski, A., Gligoroski, D.: Attacks on the Transposition Ciphers using Optimization Heuristics. In: Proceedings of ICEST 2003, pp. 1–4 (2003)

6. Russell, M., Clark, J.A., Stepney, S.: Making the most of Two Heuristics: Breaking Transposition Ciphers with Ants. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 146–147. Springer, Heidelberg (2003)
7. Garici, M.A., Drias, H.: Cryptanalysis of Substitution Ciphers Using Scatter Search. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3562, pp. 31–40. Springer, Heidelberg (2005)
8. Toemeh, R., Arumugam, S.: Breaking Transposition Cipher with Genetic Algorithm. Electronics And Electrical Engineering 7(79), 75–78 (2007)
9. Clark, J.A., Jacob, J.L., Stepney, S.: The Design of S-Boxes by Simulated Annealing, pp. 1533–1537. IEEE, Los Alamitos (2004)
10. Mao, W.: Modern Cryptography: Theory and Practice, 1st edn. Prentice Hall PTR, Englewood Cliffs (2003)
11. Ming, Z., Shu-dong, S.: Genetic Algorithms: Theory and Application. National Defense Industry Press, Beijing (2002)
12. Clark, A., Dawson, E.: Optimisation Heuristics for the Automated Cryptanalysis of Classical Ciphers. Journal of Combinatorial Mathematics and Combinatorial Computing, Papers in honour of Anne Penfold Street 28, 63–86 (1998)
13. Clark, A.J.: Optimisation Heuristics for Cryptology, PhD thesis, Queensland. University of Technology (1998)
14. Laskari, E.C., Meletiouc, G.C., Stamatioud, Y.C., Vrahatis, M.N.: Evolutionary computation based cryptanalysis: A first study, pp. 823–830. Elsevier, Amsterdam (2005)
15. Bafghi, A.G., Sadeghiyan, B.: Finding Suitable Differential Characteristics for Block Ciphers with Ant Colony Technique, pp. 418–423. IEEE, Los Alamitos (2004)
16. Ayman, E., Albassal, M.B., Abdel-Moneim, Wahdan, A.: Genetic Algorithm Cryptanalysis Of The Basic Substitution Permutation Network, pp. 471–475. IEEE, Los Alamitos (2004)

Driver Recognition Using Gaussian Mixture Models and Decision Fusion Techniques

Kristin S. Benli, Remzi Düzağaç, and M. Taner Eskil

Department of Computer Engineering,
Işık University, Şile, Istanbul, Turkey
{kristin,eskil}@isikun.edu.tr,
remzi.duzagac@isik.edu.tr
http://pi.isikun.edu.tr

Abstract. In this paper we present our research in driver recognition. The goal of this study is to investigate the performance of different classifier fusion techniques in a driver recognition scenario. We are using solely driving behavior signals such as break and accelerator pedal pressure, engine RPM, vehicle speed, steering wheel angle for identifying the driver identities. We modeled each driver using Gaussian Mixture Models, obtained posterior probabilities of identities and combined these scores using different fixed and trainable (adaptive) fusion methods. We observed error rates as low as 0.35% in recognition of 100 drivers using trainable combiners. We conclude that the fusion of multi-modal classifier results is very successful in biometric recognition of a person in a car setting.

Keywords: Recognition, Vehicle, Gaussian Mixture Model, Decision Fusion, Biometrics.

1 Introduction

Identification of a person in a vehicle setting using solely behavioral signals is a new research area in which few scientific studies were made. The research in this field has many application areas such as authentication, access control, keyless entry, secure communications and automated personalization of vehicle controls.

Specifically person recognition within a vehicle could provide the following benefits [1]:

1. Vehicle safety: Require authorization before or during the driving of the vehicle to make sure that the current driver is an authorized driver. An unauthorized driver can be denied the control of the vehicle.
2. Vehicle personalization: Adjust the vehicle controls according to the driver's physical and behavioral characteristics. Thus, a safer, more comfortable and more efficient driving environment is obtained and the distraction is minimized.
3. Secure mobile transaction opportunities: Mobile banking using biometric authentication may be an example of these opportunities.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 803–811, 2008.
© Springer-Verlag Berlin Heidelberg 2008

For person recognition, we must quantify the characteristic traits of humans and do recognition using a set of robust and discriminative criteria. In Latin, "bios" means "life" and metric means "measure". Therefore, bio-metrics is the study of methods for recognizing people according to one or more real and measurable physical or behavioral traits. Human beings recognize each other using biological characteristics like face, voice or gait. One of the well-known characteristic of the person is the fingerprint which has been used for more than hundred years for person identification. In this study biometrics-based driver identification using solely driving behavior signals is proposed.

Conventionally biometric systems focus on a single trait, such as fingerprints. However the recognition performance of the biometric systems can be improved by using multiple biometric modalities like multiple fingers, or both the face and fingerprint of a person. These systems are known as *multimodal biometric systems* and they are expected to be more reliable due to the presence of multiple pieces of evidence [2].

In this study we use the Nagoya University CIAIR database [3] for driver recognition task. Driving behavior for each driver is modeled using Gaussian Mixture Models (GMMs). In the literature GMM are frequently encountered in text-independent speaker recognition.They were used in the context of modeling the driving signals for the first time by Igarashi[4]. As Igarashi, we used the well-known Expectation Maximization algorithm for training the GMMs.

The goal of this research is to study various decision fusion techniques. We tested the performance of decision fusion using both fixed and trainable methods. Fixed methods have simple fixed rules to combine information from a set of classifiers. We made experiments using five fixed rules; Maximum, Minimum, Median, Mean and Product rules. Trainable methods have free parameters that can be trained on a separate part of the training data. In this study six trainable combiners were used; Fisher, Linear Discriminant, Nearest Mean, Perl, Parzen and K-Nearest Neighbor methods.

This paper reports the results of our study in driver recognition in a database of 100 subjects. In the next section, behavioral signals that are used in the experiments are described. Section 3 revisits the theory of classification in the context of GMMs. Section 4 introduces the decision fusion techniques that are tested in this study. Section 5 details the experiments and Section 6 presents the experimental results. We conclude with our contributions and future research directions.

2 Behavioral Driving Signals

The Center for Integrated Acoustic Information Research (CIAIR) at Nagoya University collected a multi-modal dataset of behavioral signals inside a vehicle [3]. The developers called this system as Data Collection Vehicle (DCV). During the driving process two categories of data were recorded:

– conversations of the driver subjects and
– driving behavior such as the vehicle speed, engine RPM, accelerator/brake-pedal pressure, and steering-wheel motion

In this study we only used driving behavior signals and we show that these signals are sufficient for person identification. A 100 person subset, 50 female and 50 male, of CIAIR database is used in our analysis. For each subject, there are 5 different driving behavior signals available in the CIAIR database. Each channel was sampled at 1.0 kHz and digitized to a signed 16-bit format.

1. Brake pedal pressure (kgf:kilogram force*[1]):
   0 - 50 kgf is mapped to 0 - 5.0V and linearly digitized in the range 0 - 32767
2. Accelerator pedal pressure (kgf:kilogram force*):
   0 - 30 kgf is mapped to 0 - 5.0V and linearly digitized in the range 0 - 32767
3. Engine speed (rpm:revolution per minute):
   0 - 8,000 rpm is mapped to 0 - 5.0V and linearly digitized in the range 0 - 32767
4. Vehicle speed (km/h:kilometer per hour):
   0 - 120 km/h is mapped to 0 - 5.0V and linearly digitized in the range 0 - 32767.
5. Steering angle (degree):
   -1800 degrees to +1800 degrees is mapped to -5.0 to 5.0V and linearly digitized in the range -32768 to 32767.

3 Modeling with Gaussian Mixture Probability Densities

In one dimension the Gaussian probability density function is a bell shaped curve described by two parameters that are mean μ and variance σ^2. In D-dimensional space it is described in a matrix form as

$$N(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^{D/2}\,|\Sigma|}} exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \quad (1)$$

where μ represents the mean vector and Σ represents the covariance matrix.

Gaussian distribution is usually a good approximation for a class model shape in an appropriately selected feature space. It is a mathematically sound function that could be extended to multiple dimensions. In the Gaussian distribution there is an assumption that the class model is a complete model of one basic class. It fails when the actual model, the actual probability density function, is multimodal. For example, assume that we are searching for different face parts from a picture and there are many basic types of eyes, because the people are from different races. Therefore a single Gaussian approximation model would fail to define a wide mixture of all eye types [5].

Gaussian Mixture Model (GMM) is a mixture of several Gaussian distributions. Therefore it can represent the probability distribution in a class that can

[1] 1 kilogram force = 9.80665 Newtons.

be further divided into subclasses. Such a mixture of probability density functions can be represented as a weighted sum of Gaussians

$$p(x;\theta) = \sum_{c=1}^{C} \alpha_c N(x;\mu_c,\Sigma_c) \tag{2}$$

where α_c represents the weight of the component c, $0 \prec \alpha_c \prec 1$ for all components, and $\sum_{c=1}^{C} \alpha_c = 1$. The parameter list

$$\theta = \{\alpha_1, \mu_1, \Sigma_1, ..., \alpha_c, \mu_c, \Sigma_c\} \tag{3}$$

describes a particular Gaussian mixture probability density function.

Estimation of the Gaussian mixture parameters for one class can be viewed as an unsupervised learning of those subclasses which contribute to the overall distribution. This is the case where the samples are generated by individual components of the mixture distribution and there is no knowledge about which sample was generated by which component. Clustering attempts to identify the exact components, but Gaussian mixtures can be used for modeling an aggregate distribution of unknown number of classes[5].

4 Decision Fusion

Fusion is combining information from different sources for improving the performance of a system. The most known example of fusion is the use of different sensors for detecting a target. Even if the information is coming from a single sensor, experts may interpret it differently and reach different conclusions. In this case, many experts may be "consulted" to come up with a single decision with highest confidence.

Fusion can be used for many purposes like detection, recognition, identification, tracking, decision making, etc. Information and decision fusion find application areas in defense, robotics, medicine, space and many others. Fusion processes can be categorized into low, intermediate and high level fusion according to the processing stage where the fusion takes place. Low level fusion is also called as *data fusion*. It combines raw data from the several sources for producing new raw data which is expected to be more expressive than the inputs.

Intermediate level fusion is also called as *feature level fusion*. It combines several features like edges, corners, lines, texture parameters, etc. into a feature set. The features may come from several raw data sources like several sensors or come from the same raw data. These features may be obtained from the several feature extraction methods.

High level fusion is also called *decision fusion*. It combines decisions which come from several experts. In other words, if the experts return a confidence (score) instead of a decision, it is a decision fusion problem. Voting methods, statistical methods and fuzzy logic based methods have been used in the literature for decision fusion.

Decision fusion methods can be divided into two main categories; *fixed rules* and *trainable combiners*. Fixed Rules use simple and unchangeable rules for combining different classifiers' results. Trainable combiner rules have free parameters which can be trained on a separate part of the study data. In other words, trainable combiners are classifiers themselves. These combiners do the classification in the score space instead of the feature space.

Let us assume that *S(i,j)* be the score of person i in the modality j for an input test data x. We drop x from the notation for simplicity. The goal is to get a score value *S(i)* for person i by using combination methods [6].

In this sudy, we will use the following fixed and trainable combiner methods for a comparative analysis of their performances.

Fixed Rules

1. Maximum Rule : $S(i) = max_j\ S(i,j)$
2. Minimum Rule : $S(i) = min_j\ S(i,j)$
3. Mean (Sum) Rule : $S(i) = sum_j\ S(i,j)$
4. Product Rule : $S(i) = prod_j\ S(i,j)$
5. Median Rule : $S(i) = median_j\ S(i,j)$

Trainable Combiners

1. Nearest Mean Combiner (NMC): Simple linear combiner that accepts nearest class mean as a classifier output.
2. Fisher Combiner (Fisher): Linear classifier that uses the least squares method for matching features and class labels.
3. Linear Discriminant Combiner (LDC): Linear classifier that models every class as a Gauss distribution with the same covariance matrix.
4. Parzen Combiner (Parzen): It uses Parzen density distribution function.
5. K-Nearest Neighbour Combiner (KNN): A method for classifying objects based on the classes of the closest training examples in the feature space.
6. Perl Combiner (Perl): Linear classifier by linear perceptron. A perceptron can be viewed as a binary classifier.

5 Experiments

We analyzed the performance of decision fusion methods on an in-vehicle driver recognition problem. The experiments were done using a 100 person subset of the Nagoya University CIAIR database. This subset consisted of fifty female drivers and fifty male drivers. During the experiments five different "driving behavior signals" were used.

In the first stage noise removal operation was carried out using a low pass filter. This step was followed by decimation procedure which reduces the original sampling rate for a sequence to a lower rate, which is the opposite of interpolation.

The driving signals of each driver were divided into the 20 equal length parts. First 17 parts were used for training, following 2 parts were held-out, and the final part was used for testing. Using a cross validation procedure, 20 different experiments could be carried out on a single person. The held-out data is a part of available training data that is not used during training or testing, but it is used to tune (i.e train) the decision combiner of the recognition system. Held-out data is also called *validation data*.

We used 8 mixture components for GMMs for modeling the driving signals of each person. Also, background GMMs were trained for each modality. In background model, 16 mixture (twice the number of mixtures) components were used [6]. Background GMMs were used for normalization in likelihood ratio testing for biometric recognition.

In the recognition study, the posterior probabilities of driver identities were determined for each channel in the given test data. The identity with the highest probability is chosen as the identity of the test segment. For each channel, the probability of the chosen identity is called a *score* [4]. The similarity between biometrics data is shown with these scores.

One important issue in classifier combination at the score level is to normalize the scores from each modality before fusion. Typical likelihood ranges can be very different among modalities, hence the log likelihood-ratio scores from different modalities cannot be directly added. We normalized the scores using the mean and standard deviation of likelihood scores that were calculated from the held-out validation data. Normalization can be done using a sigmoid function to map the scores to the (0,1) range.

$$S_k^{'} = \frac{1}{1 + exp(-(S_k - \mu)/\sigma)} \tag{4}$$

In equation 4, S_k represents the old log-likelihood-ratio score for the k^{th} modality and $S_k^{'}$ represents the new score. Also, μ and σ represent mean and standard deviation of old scores obtained on the validation set. After normalization these scores were combined by using fixed and trainable fusion methods. *PRTools* [7] software library is used for evaluating the results and combining the classifiers.

6 Results - Driver Recognition

The decision fusion techniques described in the previous paragraphs can be applied to driver recognition where the GMMs trained for each modality suggest a decision. As a result, we have decisions for each modality and these decisions need to be combined in a way that maximizes the performance. This section will present our findings in the driver recognition problem.

Table 1 presents the performance of driver recognition experiments using the individual modalities (i.e. without decision fusion). The best experiment result was obtained with the accelerator pedal presure.

Table 1. Individual performance results for different modalities

Modality	Percent Error
Brake	90.35
Accel	86.15
RPM	98.05
Speed	97.35
Steering	97.90

The highest error rate was obtained with engine speed [2]. The results shown on Table 1 imply that individual driving signals are not appropriate for biometric identification.

This result is compatible with the findings of Wakita et. al. [8]. They present 3 possible reasons for accelerator pedal pressure preserving the most personal property information:

- The accelerator pedal pressure is the direct result of force exerted by the driver
- It is used more frequently than other controls, such as the break pedal
- The acceleration of the vehicle conveys the driving taste of the operator, whereas other signals (e.g. break pedal pressure or steering wheel angle) are more dependent on the driving conditions.

The next stage of our research is driver recognition using the fixed rules and trainable combiners. In this study four different combinations of modalities were chosen and experiments were done using five different fixed rules and six different trainable combiners.

Fixed rules do not show good performances. Their error rates are very high. The lowest error rate, 77.95 percent, was obtained on Combination 1[3] using the product rule. This error rate is the best we could achieve using fixed rules. Moreover, product rule shows the best performance in all the experiments and has the lowest error rates. Mean rule follows the product rule with the second best lowest error rates.

Overall, the best experiment results were obtained using trainable combiners and specifically the Parzen and K-NN methods. These methods reached error rates as low as 0.35%. Fisher linear combiner showed good performances when

[2] Abbreviations are used for different channels. These channels are break pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle respectively.

[3] Note: In the combine list, comma (,) shows the decision fusion where the posterior probabilities of classifiers are combined.
Comb 1 = Brake, Accelerator
Comb 2 = Brake, Accelerator, RPM, Speed, Steering
Comb 3 = Brake, Accelerator, Steering
Comb 4 = Brake, Accelerator, RPM, Speed

Table 2. Percent error rates using fixed rules and trainable combiners

Methods	Comb1	Comb2	Comb3	Comb4
Max	85.95	91.60	89.30	90.75
Min	81.40	90.55	86.50	90.50
Median	78.95	87.70	86.45	86.10
Mean	78.95	85.95	82.30	85.35
Prod	77.95	84.85	81.20	85.05
Fisher	43.45	16.10	29.50	24.55
LDC	35.50	85.55	48.00	78.70
NMC	66.65	54.65	61.40	60.40
Perl	66.00	38.70	55.05	44.25
Parzen	0.55	0.35	0.65	0.80
KNN	0.55	0.35	0.65	0.75

the number of input classifiers are increased such as in the second and fourth combination. Linear Discriminant combiner acted oppositely; it showed better performance when the number of input classifiers are decreased. Nearest Mean combiner is a very robust method but generally higher error rates were obtained during the experiments. Specifically in the first and third combination it reached the highest error rates.

7 Conclusion and Future Work

The ultimate goal of our research is to facilitate vehicle-person interaction. In this paper we present our results in driver recognition using solely driver behavior signals and decision fusion. We carried out driver recognition experiments using a 100 person subset of the CIAIR database. This subset consisted of 50 female drivers and 50 male drivers.

In our experiments, the driving signals such as brake pedal pressure, accelerator pedal pressure, engine speed, vehicle speed and steering wheel angle constitute the feature vectors that define the numerical characterization of the biometrics. We used GMMs for approximating the probability distribution of each of these features for each subject in our database. We trained the GMMs using the Expectation Maximization algorithm. For an input pattern we obtain the similarity scores and then combine the decisions of different "GMM experts" for improving the performance of the recognition system. We tested the performance of several fixed and trainable decision fusion techniques.

As expected, trainable combiners showed much higher performance than fixed rules. Among the fixed rules, the product rule showed slightly higher performance. The mean rule consistently rated as the second best classifier after product rule.

In general, the performance obtained with fixed rules is slightly higher than that of individual channel performances.

Among the trainable combiners, the Parzen and K-NN consistently rated better than the others. In Parzen and K-NN, we achieved the best performance (0.35%) when we used all driving signals. We obtained next better performance using the brake pedal and accelarator pedal pressures and again the Parzen and K-NN methods. These figures are very promising and they imply that in-car recognition is possible using solely the actions of the driver, specifically the usage of controller pedals. These results also point out a potential of application in a driver verification scenario in which a person who is not confirmed to be the claimed identity will be denied the controls of the vehicle.

We plan to extend our research to the driver verification task. We also plan to extend our experiments to all 812 driver subjects in the CIAIR database. This has proven to be challenging both in terms of the time and memory required for the training of the GMMs. In a parallel study we plan to apply feature reduction/extraction to driving signals to obtain the best meaningful feature set for the characterization of the driver behavior. This will also give us an opportunity to do a comparative study on intermediate (feature) level and high (decision) level fusion.

References

1. Erdogan, H., Ercil, A., Ekenel, H.K., Bilgin, S.Y., Eden, I., Kirisci, M., Abut, H.: Multi-modal person recognition for vehicular applications. In: Oza, N.C., et al. (eds.) MCS 2005. LNCS, vol. 3541, pp. 366–375. Springer, Heidelberg (2005)
2. Jain, A.K., Ross, A., Prabhakar, S.: An Introduction to Biometric Recognition. IEEE Transactions on Circuits and Systems for Video Technology 14(1) (January 2004); Special Issue on Image- and Video-Based Biometrics
3. Kawaguchi, N., Matsubara, S., Takeda, K., Itakura, F.: CIAIR In-Car Speech Corpus: Influence of Driving Status. IEICE Transactions on Information and Systems E88-D(3) (March 2005)
4. Igarashi, K., Miyajima, C., Itou, K., Takeda, K., Abut, H., Itakura, F.: Biometric Identification Using Driving Behavioral Signals. In: Proceedings IEEE ICME 2004, Taipei, Taiwan, June 27-30 (2004)
5. Paalanen, P.: Bayesian Classification using Gaussian Mixture Model and EM Estimation:Implementations and Comparisons. Lappeenranta University of Technology, Department of Information Technology, Lappeenranta, Finland, Tech. Rep. (2004)
6. Erdogan, H., Ozyagci, A.N., Eskil, T., Rodoper, M., Ercil, A., Abut, H.: Experiments on Decision Fusion for Driver Recognition. In: Biennial on DSP for In-Vehicle and Mobile Systems, Taipei, Taiwan, June 27-30 (2004)
7. Duin, R.P.W., Juszczak, P., de Ridder, D., Paclik, P., Pekalska, E., Tax, D.M.J.: PRTools, a Matlab toolbox for pattern recognition (2004), `http://www.prtools.org`
8. Wakita, T., Ozawa, K., Miyajima, C., Igarashi, K., Itou, K., Takeda, K., Itakura, F.: Driver Identification Using Driving Behavior Signals. IEICE Trans. Inf. & Syst. E89-D(3) (March 2006)

Fast Partial Distortion Elimination Algorithm Using Probability Improvement of Sub-block Uniform Distribution

TaeKyung Ryu, KwangSeok Moon, and JongNam Kim

Div. of Electronic Computer and Telecommunication Eng., Pukyong Nat'l University
{toydev,moonks,jongnam}@pknu.ac.kr

Abstract. Full search motion estimation in real-time video coding requires large amount of computations. Reducing computational cost for full search motion estimation is critical research issue for enabling fast real-time video coding such as MPEG-4 advanced video coding. In this paper, we propose a fast partial distortion algorithm using probability improvement of sub-block uniform distribution. Our algorithm uses normalized dithering matching scan to get uniform distribution of partial distortion, and based on normalized dithering order matching scan and calibration of threshold error using LOG value for each sub-block continuously for efficient elimination of unlike candidate blocks fast while keeping the same prediction quality compared with the full search algorithm. Our algorithm reduces about 65% of computations for block matching error compared with conventional PDE(partial distortion elimination) algorithm without any prediction quality.

1 Introduction

In MPEG-2 and MPEG-4 video compression, full search (FS) algorithm based on block matching algorithm (BMA) finds them optimal motion vectors which minimize the matching difference between reference block and candidate block. It has been widely used in video coding applications because of its simple and easy hardware implementation. However, heavy computational load of the full search with very large search range can be a significant problem in real-time video coding application. Many fast motion estimation algorithms to reduce the computational load of the full search have been studied in the last decades.

Several fast motion estimation algorithms have been studied in recent years in order to reduce the computational cost required. These algorithms can be classified into two main groups. One group of algorithms is based on lossy motion estimation technique with degradation of prediction quality compared with the conventional FS algorithm. The other group of algorithms is based on lossless estimation technique that does not degrade the prediction quality. The lossy group of algorithms includes unimodal error surface assumption algorithm, multi-resolution algorithm, variable search range algorithm with spatial/temporal correlation of the motion vectors, half-stop algorithm using threshold of matching distortion, and others [1]. Lossy group of algorithms includes TSS (three-step search), DS (diamond search), HEXBS (hexagon

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 812–820, 2008.
© Springer-Verlag Berlin Heidelberg 2008

based search), and others [2]-[3]. The latter as fast full search technique contains following several algorithms: successive elimination algorithm (SEA) and its modified algorithms [4]-[8], partial distortion elimination (PDE) method and its modified algorithms [9]-[14], and so on. In lossless motion estimation algorithms, PDE is very efficient algorithm to reduce unnecessary computation for matching error calculation. To further reduce unnecessary computation in calculating matching error, J.N. Kim and et al. proposed fast PDE algorithms based on adaptive matching scan, which requires additional computation to get matching scan order [12]-[13]. But the additional computation for the matching scan order can be burden when cascading other fast motion estimation algorithm such as SEA. In this paper, we propose a fast motion estimation algorithm to reduce computational load of the FS algorithm. We reduced only unnecessary computations which doesn't affect predicted images from the motion vector. To do that, we use normalized dithering matching scan to get uniform distribution of partial distortion which can reduce only unnecessary computation significantly. Additionally, to reduce prediction error in our algorithm, we add constant factor to partial $SAD_{\min}$ for prediction error reduction. Our algorithm reduces about 65% of computations for block matching error compared with the conventional PDE algorithm without any degradation of prediction quality. This paper is organized as follows. In Section 2, conventional fast ME algorithms are described. Then, Section 3 explains the motivation of them proposed algorithm based on the previous works. In Section 4, experimental results for various test sequences and discussions of the results are presented. The conclusion is followed in Section 5.

2 Conventional Works

2.1 PDE (Partial Distortion Elimination)

PDE algorithm uses the partial sum of matching distortion to eliminate impossible candidates before completing calculation of matching distortion in a matching block. That is, if an intermediate sum of matching error is larger than the minimum value of matching error at that time, the remaining computations for matching errors is abandoned. The *k*th partial sum of absolute differences (SAD) can be expressed by the Eq. (1),

$$\sum_{i=1}^{k}\sum_{j=1}^{N}\left|f_{t+1}(i,j)-f_{t}(i+x,j+y)\right| \qquad k=1,2,...N \tag{1}$$

where N represents matching block size. The term, $f_{t+1}(i,j)$, means image intensity at the position (i,j) of the $(t+1)$th frame. The variables x and y are the pixel coordinate of a candidate vector. If the partial sum of matching distortion exceeds the current minimum matching error at k, then we can abandon the remaining calculation of matching error ($k+1$ to Nth rows) by assuring that the checking point is an impossible candidate for the optimal motion vector. Kim et al. calculated block matching errors to reduce unnecessary calculations with the four-directional scan order based on the gradient magnitude of images instead of the conventional top-to-bottom matching scan order. Block matching errors are calculated to further reduce unnecessary computations with adaptive matching scan. While these approaches could reduce unnecessary

computations for getting block matching errors, they need additional computations to determine the matching scan order.

2.2 NPDS (Normalized Partial Distortion Search)

NPDS[10] algorithm adapted instead of calculating the total distortion consisting of all pixels of a 16×16 MB at one time, a grouping method is applied to divide pixels in one MB into 16 groups with evenly distributed patterns as shown in Fig. 1. As result the SAD between current MB and a candidate MB is subdivided into 16 partial distortions d_p,

$$d_p = \sum_{k=1}^{3}\sum_{l=1}^{3}\left|\begin{matrix} f_t(i+4k+s_p, j+4l+t_p) \\ - f_{t-1}(i+4k+s_p+x, j+4l+t_p+y)\end{matrix}\right| \tag{2}$$

Where the pth partial distortion is given as where $1 \le p \le 16$, and the pth accumulated partial distortion SAD_p is defined as

$$SAD_p \ge \sum_{i}^{p} d_i \tag{3}$$

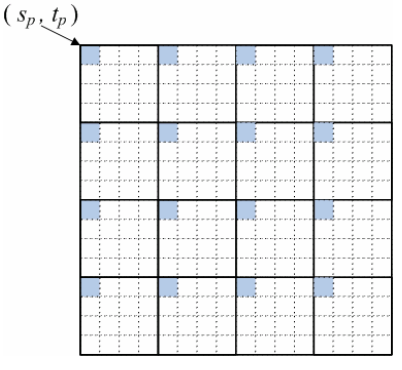


Fig. 1. Pixel grouping pattern for calculation of the partial distortion in NPDS

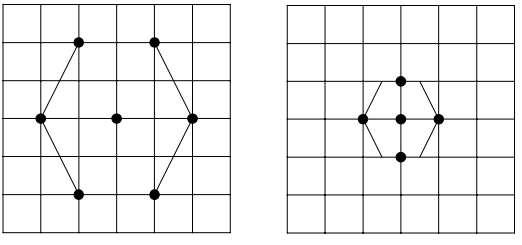
Note that, in the above equations, $f_t(x,y)$ and $f_{t-1}(x,y)$ denote the pixel values in the current frame and reference frame, respectively. (x,y) denotes the candidate motion vector, and (s_p,t_p) indicates the offsets of the upper left corner point of the pth partial distortion from the upper left corner of the MB. The order of calculation of the 16 partial distortions is illustrated as in the upper left part of Fig. 1. In Eq. (4). the pth partial SAD to check during the matching is as follows.

$$SAD_p \ge p \cdot \frac{1}{N} SAD_{min} \tag{4}$$

2.3 HEXBS (Hexagon Based Search)

The HEXBS algorithm uses a large hexagon-shaped pattern (LHSP) and a small hexagon-shaped pattern (SHSP), as depicted in Fig. 2. A large hexagon-shaped search

pattern is depicted in Fig. 2(a), which consists of seven checking points with the center surrounded by six end points of the hexagon with distance 2 and $\sqrt{5}$ from the center point, respectively Of the six endpoints in the hexagon, two horizontal points are away from the center with distance 2 and the remaining four points have a distance of from the center point, respectively. From the figure, we can see the six end points are approximately uniformly distributed around the center. In the search process, the large hexagon-shaped search pattern keeps advancing with the center moving to any of the six end points. Whichever end point the center of the search pattern moves to, there are always three new end points emerging, and the other three endpoints are being overlapped. Fig. 2(b) illustrates a smaller shrunk hexagonal pattern covering four checking points (left, right, up, and down dots around the center with distance 1) in the motion field, which is finally used in the focused inner search.



(a)LHSP (b)SHSP

Fig. 2. Hexagon Search Pattern

It searches fewer points in the horizontal direction shift and it is unable to converge completely in the final step due to the less searching points. These factors cause the predict quality of HEXBS.

3 Proposed Algorithm

Modified PDE algorithms have been published by using adjacent motion vectors in the spatial or temporal domain, spiral scan algorithm and cascaded algorithms to other fast ones [13]-[14]. Ability to reject impossible candidates in the PDE algorithm depends on the search strategy, which makes minimum matching error be detected faster. For the purpose, spiral search is very efficient. PDE algorithm with spiral search rejects impossible candidates faster than simple PDE. Therefore, we employ the spiral search in the proposing matching scan algorithms.

To improve matching scan strategy of NPDS, we need more efficient matching scan order of uniform error distribution for each sub-block. Our proposed algorithm improved sub-block matching strategy using 4×4 normalized dithering matching scan order to spread sub-block error distribution uniformly. Fig. 3 shows dithering matching order in 4×4 matrix and pixel grouping pattern in each MB. Eq. (5) shows proposed SAD values for each sub-block is as follows. In Eq. (5), (x,y) denotes the

candidate motion vector, and (ds_p, dt_p) indicates the offsets of the upper left corner point of the pth partial distortion from the upper left corner of the MB.

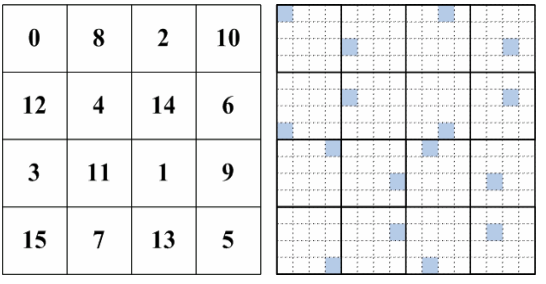


Fig. 3. Proposed pixel grouping pattern using dithering matrix

$$d_p = \sum_{k=1}^{3}\sum_{l=1}^{3}\left| f_t(i+4k+ds_p, j+4l+dt_p) - f_{t-1}(i+4k+ds_p+x, j+4l+dt_p+y) \right| \tag{5}$$

Our algorithm also doesn't have ideally distributed sub-block matching error but improve uniform distribution for each block. Distribution error mostly occurred in the first sub-block matching case. To reduce prediction error in our algorithm, we must multiply or add factor to partial $SAD_{\min}$. Therefore, we consider the probability of max error distribution for every MB in each test videos, and find multiply factor between pth accumulated SAD_p and partial $SAD_{\min}$ comparison for each sub-block.

Fig. 4 shows accumulated error probability of 100 frames of 'foreman', 'trevor', 'claire' and 'grand mother' video sequences. In these sequences, 'foreman', has higher motion variance than the other image sequences. We consider 98% of accumulated error factor for 'foreman' video sequence, take the factor value 1.7 is equal to

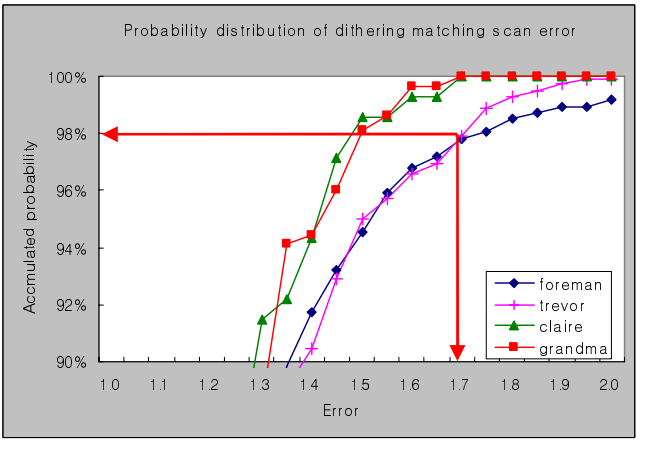


Fig. 4. Probability distribution of dithering matching scan

$\log_6(6+N\text{-}1)$. We compare it with factor with log value ($\log_{base}(base+N\text{-}1)$) in Eq. (6). The meaning of the symbols in Eq. (6) is when N is 16, the factor comes to 1.

$$SAD_p \geq p \times \frac{1}{N} \times SAD_{min} \times \log_{base}(base + N - p) \tag{6}$$

4 Experimental Results and Discussions

To compare the performance of the proposed algorithm with the conventional algorithms, we use 100 frames of 'foreman', 'car phone', 'trevor', 'claire', 'akio' and 'grand mother' image sequences. In these sequences, 'foreman', and 'car phone' have higher motion variance than the other image sequences. 'claire', 'akio' and 'grand mother' are rather inactive sequences compared with the first two sequences. 'trevor' sequence has intermediate level of motion variance. Matching block size is 16×16 pixels and the search window is ± 7 pixels. Image format is QCIF (176×144) for each sequence and only forward prediction is used.

The experimental results shown in Fig. 5 and 6 and Table 1 and 2 are presented in terms of average number of checking rows with reference to that of full search without any fast operation. Table 3 and 4 show the differences of the peak-to-peak-signal-to-noise ratio (PSNR) performance of the proposed algorithm and FS and HEXBS. All the algorithms employed spiral search scheme to make use of the distribution of motion vectors.

Fig. 5 and 6 show the reduced computation of average checking rows based on PDE algorithm. The proposed uniformly distributed sub-block matching scan algorithm significantly reduces unnecessary calculations compared with the conventional

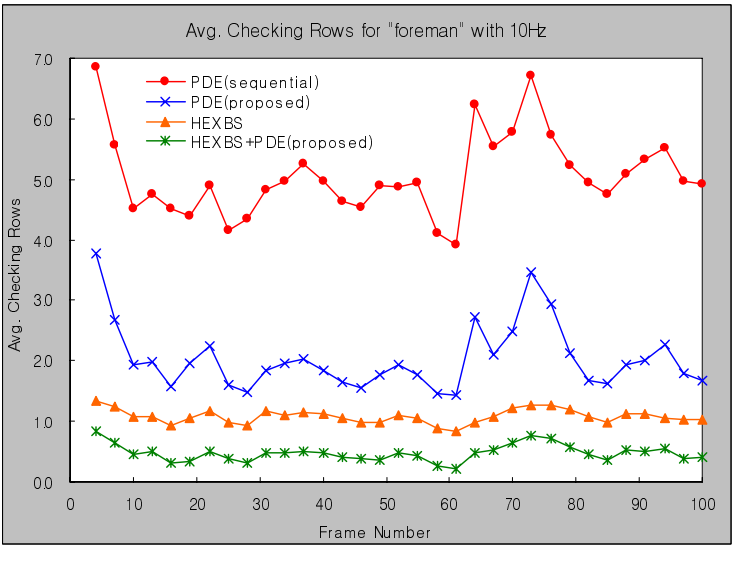


Fig. 5. Average number of rows computed for "foreman" sequence of 10Hz

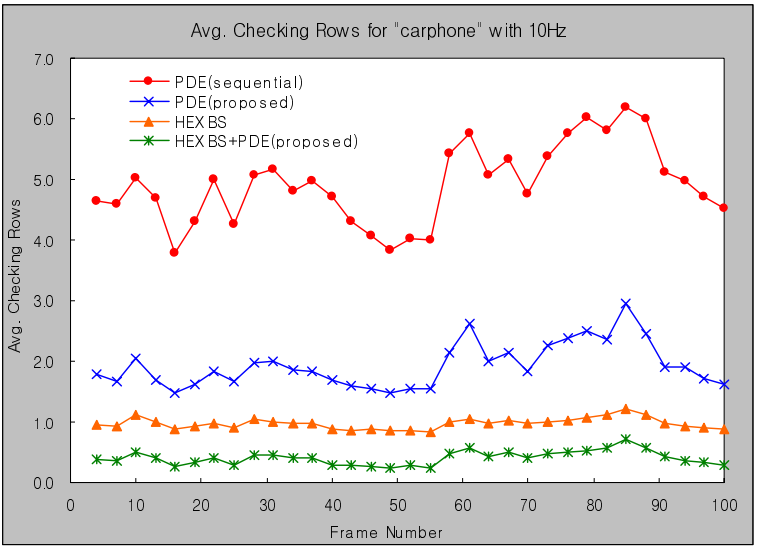


Fig. 6. Average number of rows computed for "car phone" sequence of 10Hz

sequential scan algorithm. We apply the proposed algorithm to HEXBS to further demonstrate the performance. From the experimental results, we can see that the proposed matching algorithm can reduce unnecessary computations efficiently for PDE itself and cascaded algorithms with HEXBS. Table 1 and Table 2 summarize average numbers of checking rows computed for various algorithms in all sequences of 30Hz and 10Hz. The average number of checking rows of the conventional full search algorithm without any fast operation is 16. From the Table 1 and Table 2, we can see that the computational reduction ratios from the proposed matching scan combined with HEXBS is average 72% compared with the conventional HEXBS. The results of PSNR differences in Table 3 and Table 4 are regardless to consider.

With the experimental results, we can conclude that our uniformed matching scan algorithm can reduce computational cost significantly without any degradation of prediction quality and any additional computational cost for obtaining matching order.

Table 1. Average numbers of rows for all sequences for 30Hz

Algorithms	Fore-man	Car phone	Trevor	Claire	Akio	Grand
Original FS	16.00	16.00	16.00	16.00	16.00	16.00
PDE (sequential)	4.03	4.21	3.13	3.88	1.63	4.11
PDE (proposed)	1.44	1.60	1.29	1.20	1.08	1.93
HEXBS	0.90	0.89	0.85	0.83	0.83	0.85
HEXBS+PDE(proposed)	0.26	0.29	0.20	0.19	0.14	0.24

Table 2. Average numbers of rows for all sequences for 10Hz

Algorithms	Fore-man	Car phone	Trevor	Claire	Akio	Grand
Original FS	16.00	16.00	16.00	16.00	16.00	16.00
PDE (sequential)	5.05	4.91	4.51	4.40	2.02	4.61
PDE (proposed)	2.04	1.93	1.83	1.28	1.11	1.98
HEXBS	1.08	0.97	0.93	0.84	0.84	0.86
HEXBS+PDE(proposed)	0.48	0.41	0.34	0.23	0.15	0.28

Table 3. PSNR difference between FS and proposed PDE algorithms at 30Hz and 10Hz

	Foreman	Car phone	Trevor	Claire	Akio	Grand
30Hz	0.002	0.003	-0.002	0.001	0.000	0.001
10Hz	0.004	-0.003	0.007	0.002	0.001	-0.006

Table 4. PSNR difference between HEXBS and proposed PDE algorithms with HEXBS at 30Hz and 10Hz

	Foreman	Car phone	Trevor	Claire	Akio	Grand
30Hz	0.000	0.002	0.004	0.001	0.000	0.001
10Hz	0.020	0.007	0.004	0.003	0.001	-0.005

4 Conclusions

In this paper, we propose improved sub-block matching scan algorithm using normalized dithering matching scan which can reduce only unnecessary computation significantly. The proposed is based on normalized dithering order matching scan and calibration of threshold error using LOG value for each sub-block continuously for efficient elimination of unlike candidate blocks while keeping the same prediction quality compared with conventional full search algorithm. The proposed algorithm can be efficiently cascaded to other fast algorithms such as HEXBS. Experimental results show that the proposed algorithm reduces 65% of computations for block matching error compared with PDE algorithm without any prediction quality. The proposed algorithm will be particularly useful for realizing fast real-time video coding applications, such as MPEG-4 advanced video coding, that require large amount of computations.

Acknowledgements

This work was supported from Advanced Technology Project by SMBA and RIS Project by KOTEF.

References

1. Dufaus, F., Moscheni, F.: Motion estimation techniques for digital TV: A review and a new contribution. IEEE Proceeding 83, 858–876 (1995)
2. Zhu, S., Ma, K.K.: A new diamond search algorithm for fast block-matching motion estimation. IEEE Trans. Image Processing 9(5), 287–290 (2000)
3. Zhu, C., Lin, X., Chau, L.P.: Hexagon-based search pattern for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol. 12(5), 349–355 (2002)
4. Li, W., Salari, E.: Successive elimination algorithm for motion estimation. IEEE Trans. Image Processing 4, 105–107 (1995)
5. Lu, J.Y., Wu, K.S., Lin, J.C.: Fast full search in motion estimation by hierarchical use of Minkowski's inequality (HUMI). Pattern Recog. 31, 945–952 (1998)
6. Coban, M.Z., Mersereau, R.M.: A fast exhaustive search algorithm for rate-constrained motion estimation. IEEE Trans. Image Processing 7, 769–773 (1998)
7. Wang, H.S., Mersereau, R.M.: Fast algorithms for the estimation of motion vectors. IEEE Trans. Image Processing 8, 435–438 (1999)
8. Gao, X.Q., Duanmu, C.J., Zou, C.R.: A multilevel successive elimination algorithm for block matching motion estimation. IEEE Trans. Image Processing 9, 501–504 (2000)
9. Ahn, T.G., Moon, Y.H., Kim, J.H.: Fast full-search motion estimation based on Multilevel Successive Elimination Algorithm. IEEE Trans. Circuits System for Video Technology 14, 1265–1269 (2004)
10. Cheung, C.K., Po, L.M.: Normalized partial distortion search algorithm for block motion estimation. IEEE Trans. Circuits Syst. Video Technol. 10, 417–422 (2000)
11. Kim, J.N.: Adaptive matching scan algorithm based on gradient magnitude for fast full search in motion estimation. IEEE Trans. Consumer Electronics 45, 762–772 (1999)
12. Kim, J.N., Byun, S.C., Kim, Y.H., Ahn, B.H.: Fast Full Search Motion Estimation Algorithm Using Early Detection of Impossible Candidate Vectors. IEEE Trans. Signal Processing 50, 2355–2365 (2002)
13. Kim, J.N., Ryu, T.K., Jeong, Y.J.: A Fast Partial Distortion Elimination Algorithm Using Selective Matching Scan. In: Levi, A., Savaş, E., Yenigün, H., Balcısoy, S., Saygın, Y. (eds.) ISCIS 2006. LNCS, vol. 4263, pp. 125–133. Springer, Heidelberg (2006)
14. Ryu, T.K., Jeong, T.I., Ryu, K.Y., Moon, K.S., Kim, J.N.: A Fast Partial Distortion Elimination Algorithm Using Adaptive Matching Scan and Refining Threshold. In: International Conference on Intelligent Computing (ICIC 2007), pp. 801–809 (August 2007)

PSO Combined with ILS for Flowshop-Based Parallel Task Assignment Problem

Runpeng Liang, Jiaxiang Luo, Qingqiang Yang, and Wenfeng Luo

Ministry of Education Engineering Center of Sophisticated Electronic Equipment Manufacturing, South China University of Technology, Guangzhou 510641, China
luojx@scut.edu.cn

Abstract. This research explores a new scheduling problem in electronics mounting assembly line: flowshop-based parallel task assignment (FSBPTA), in which the machines are lined flowingly and the tasks—including both the allocation of PCBs and their components, will be assigned to machines for mounting with the objective of minimizing the makespan. The allocation of PCBs has the characteristic of the flowshop scheduling problem (FSSP) while the assignment of the components resembles the job assignment in parallel-machine scheduling problem. To solve this problem, a hybrid algorithm method of Particle Swarm Optimization (PSO) and Iterated Local Search (ILS) is proposed. In this algorithm, ILS is used to help PSO algorithm escape from being trapped into local optima. To testify this hybrid algorithm (PSO&ILS), comparisons between PSO&ILS and PSO, PSO&ILS and multi-start descent algorithm have been advanced. Results show a 9.80% and 3.70% improvement respectively, which illustrate the effectiveness of the proposed algorithm.

Keywords: Hybrid algorithm, Particle Swarm Optimization, ILS, Parallel task assignment.

1 Introduction

Electronics assembly manufacturing, which involves the mounting of no-pin or short-lead devices onto the printed circuit boards (PCB), is a continuous and high-speed mass production workshop that deals with quantities of multi-category small components. Recently, due to the increasingly intense competition in the market, these manufacturing enterprises have been attempting to lower their costs by increasing the efficiency of the assembly line, especially by means of improving the efficiency of its core equipment—mounting machine. In view of its huge costs, mounting machine has become the bottleneck equipment of the electronics assembly manufacturing and a breakthrough in an effective scheduling or approach may revolutionize the productivity of the enterprises in terms of reducing the idle waiting time of the machines and improving the utilization of the resources. In consideration of its engineering significance, the research on the efficiency of the mounting machines will not only be well-deserved but also necessary.

Under most circumstances, components on the PCBs are operated by one of the flowingly lined mounting machines and the machines are usually classified into

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 821–830, 2008.
© Springer-Verlag Berlin Heidelberg 2008

different types, which process different kinds of components with different precision and processing time. In this case, researches on the permutation of the PCBs and the allocation of the components onto different machines may be feasible and effective in optimizing the mounting time. Based on this rationale, this article focuses on the sequencing of the PCBs and their corresponding components, and we call this novel scheduling problem flowshop-based parallel task assignment (FSBPTA). The illustration of FSBPTA can be shown briefly in Fig.1:

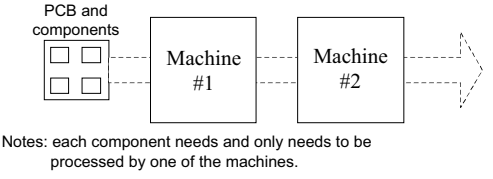


Fig. 1. Flowshop-based parallel task assignment

In general, the problem can be described as follows: there are a set of m independent PCBs with n_i components (i=1,2…m) on them respectively that need to be processed through a set of k machines. The processing time of each component on each machine is roughly assumed to be a fixed number p_{ijk}, where i,j,k are integers and stand for the index number of PCB, component, machine respectively. A machine processes one PCB at a time exclusively and one PCB is conveyed from one machine to another without preemption. An operation of PCB n+1 starts to be processed only if the previous operation of PCB n is completed. Here, there is no buffer between any two machines. The objective of this problem is to find the optimal sequence of both the PCBs and components in order to minimize the makespan.

Also, according to the characteristic of FSBPTA, for component a_{ij}, if it is arranged to be processed on machine k, then we will have p_{ijk}>0, which means it will pass other machine l (l≠k) without any operation (p_{ijk}=0). To simplify this problem, this article assumes that there are only two mounting machines available for the PCB processing. This assumption does not undermine the value of the research since parallel-machine scheduling problem with its special case m=1 is already NP-hard in the strong sense [1].

From the above description, it can be seen that FSBPTA has both the characteristic of FSSP and that of parallel-machine scheduling problem. On the one hand, like the jobs in FSSP, every PCB and its corresponding components need to be conveyed from the first machine to the last machine without preemption. On the other hand, unlike FSSP, each component needs and only needs to be operated by only one of the machines, which is the characteristic of parallel-machine scheduling problem. With the combination of FSSP and parallel-machine scheduling problem, we are sure FSBPTA has no less research value than either of the classical problem mentioned above.

To solve this problem, we can first refer to the methods that applied to tackle the classical FSSP and parallel-machine scheduling problem. For the past few years, heuristic algorithms have been applied to scheduling problems due to their effectiveness in solving NP-hard problems and making a trade-off between solution quality

and computation time. For example, genetic algorithm, ant-colony algorithm, tabu search algorithm and other heuristic algorithms have been applied to tackle FSSP and gained satisfactory results. Also, extensive and growing series of heuristic methods have seen their success in solving parallel-machine scheduling problem.

As one of the heuristic algorithms, PSO is a population-based evolutionary algorithm that has a wide variety of applications for solving scheduling problems [2]. In view of its satisfactory results in solving both FSSP [3] and parallel machine scheduling problem [4] as well as in consideration of its computational feasibility and simplicity, this article attempts to tackle FSBPTA with the aid of PSO. However, in view of the particles' early convergence during the updating process, this paper combines PSO with ILS by incorporating basic swap and insert strategies to further explore the global best solution of PSO and applying kick strategy to the solution when it is trapped into local optima, which guarantees the diversity of the solutions and makes it possible to improve the solution quality. Computational results found later in this article lend support to our claims about the improvement.

The remainder of the paper is organized as follows: the introduction of PSO is presented in the next section. In section 3, we try to introduce PSO into the solving of FSBPTA and the construction of PSO&ILS is explained in details. Section 4 shows the computational results and the corresponding conclusions. In section 5, the effectiveness of the proposed PSO&ILS is evaluated.

2 Brief Introduction of PSO

PSO was developed by Kennedy and Eberhart, by means of simulating the movement of the bird flocks. Like other evolutionary algorithm, PSO formulates a multidimensional search space to explore solutions. Each position in the search space stands for one feasible solution to the problem. The status of each particle is characterized by its position and velocity. After initialization, particles are treated like a point in the search space and they evolve from iteration to iteration by updating their own position and velocity. In order to avoid the randomness and blindness of the particles' movement, an updating mechanism that incorporates the information sharing of the best position of the particles and the best position ever found by the whole swarm is presented in the frame of PSO. This sharing mechanism enables the particles to have memories and thus they can adjust their movement toward the best position in an effective way. The updating mechanism can be described by the following equations:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_1[pb_{i,j} - x_{i,j}(t)] + c_2 r_2[pb_{g,j} - x_{i,j}(t)] \quad . \tag{1}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad . \tag{2}$$

In the above equations, $x_{i,j}(t)$, $v_{i,j}(t)$ respectively represent the position and velocity of particle i on dimension j at iteration t. $pb_{g,j}$ is the best position of the swarm on dimension j while $pb_{i,j}$ is the best position of particle i on dimension j. w is called inertia weight and the particles obtain high velocities with a large w while obtain low velocities with a small one. Constants c_1 and c_2 are called learning factors and they are used to determine whether particles move toward $pb_{i,j}$ or $pb_{g,j}$. r_1 and r_2 are

random variables and uniformly distributed between 0 and 1. For particles, there is one boundary for their movement: particle position should satisfy $x_{min} \le x_{ij} \le x_{max}$ and velocity should meet the relation $v_{min} \le v_{ij} \le v_{max}$. Whenever x_{ij} exceeds x_{max}, x_{ij} will be replaced by x_{max}. This is the same for the instance of x_{ij} exceeding x_{min}. The same rule is applied to v_{ij}.

To apply PSO to solve a problem, the conversion from particles' movement to the actual problem, and that from continuous fields used in PSO to discrete fields of the real problems should be taken into account. Also, a fitness function should be constructed to evaluate the particles' quality.

3 PSO&ILS Applied to FSBPTA

3.1 Solution Representation

Since the makespan of all the tasks depends on the permutation of the PCBs and components, it is feasible to express the tentative solutions in the form of permutation sets. Also, since every PCB contains lots of components, and the change of the PCB's permutation will directly affect the arrangement of the components, this article attempts to reveal this intrinsic relation by establishing two kinds of particles: PCB particles and their sub-particles—component particles. Under this hypothesis, the solution can be expressed by the permutation of the PCBs as follows: $I=\{I(1)\ I(2)\ldots I(m)\}$. For PCB Z_i ($Z_i \in [1,m]$ and i represents the position in permutation set I), its solution set $I(i)$ can be expressed by its sub-particles' permutation: $I(i)=\{\phi_{Z_i 1}\ \phi_{Z_i 2}\ldots\phi_{Z_i k}\}$, where $\phi_{Z_i j}$ ($j \le k$) means the components of PCB Z_i that need to be processed on machine j. For example, if $\phi_{Z_i j}=\{\delta_1\ \delta_2\ \delta_3\ \delta_4\}$, it means components $\delta_1\ \delta_2\ \delta_3\ \delta_4$ on PCB Z_i are arranged to be processed on machine j. For the rest of the components on PCB Z_i, they will be distributed to machines other than j for operation.

Under the assumption of two mounting machines in this article, the PCB particles can be expressed in the same way as aforementioned. However, when it comes to the component particles, in order to distinguish the components that operated on the first machine and the second one, one dummy 0 component is introduced into every PCB. Accordingly, the subset of $I(i)$ will be modified as $I(i)=\{\phi_{Z_i 1}\ 0\ \phi_{Z_i 2}\}$. Then if $I(i)=\{\delta_1\ \delta_2\ \ldots\delta_{nZ_i-2}\ 0\ \delta_{nZ_i-1}\ \delta_{nZ_i}\}$, it means components $\delta_1\ \delta_2\ \ldots\delta_{nZ_i-2}$ are arranged on the first machine while the rest of the components $\delta_{nZ_i-1}\ \delta_{nZ_i}$ are distributed onto the second machine.

For these two kinds of particles, they have their own updating mechanism in the form of equation (1) and (2). For every dimension j, we first update the PCB particles and search for the individual best and global best position. Second, once the permutation of the PCB particles is confirmed, we update those component particles according to the individual and global best information provided by the PCB particles.

3.2 Conversion from Particle Position to Task Sequence

Since the particles' positions are continuous values while the task sequences are discrete numbers, in order to combine the particles' position values with the actual task

allocation, we need a conversion rule that help uniquely convert the particles' position values to task arrangements. Thus, the smallest position value (SPV) rule [5] is introduced in this article for the conversion.

The fundamental idea of the SPV rule is to sort the tasks on ascending orders according to the particles' position values. For example,

Table 1. Illustration of SPV rule

Position value	Index number	Task sequence
1.7	0	1
2.1	1	2
3.5	2	4
0.4	3	0
3.2	4	3

According to SPV rule, position value 0.4 is the smallest. Consequently, particle with index number 3 is attached 0 for the real task sequence. Under this hypothesis, the suggested solution for the problem is: component 1, 2, 4 should be arranged on the first machine while component 3 should be on the second machine. All in all, this task sequence indicates one tentative solution to the problem under the mechanism of PSO.

3.3 Fitness Value of the Particles

The objective of this problem is to search the solution space for a solution that can reduce the makespan to the largest extent within an acceptable time. According to the aforementioned hypothesis, for the solution set $I=\{I(1)\ I(2)\ldots I(m)\}$, the makespan of the two machine FSBPTA problem can be expressed as follows:

$$T_{total} = T(\phi_{Z_{11}}) + \sum_{j=1}^{m-1} \max(T(\phi_{Z_{j2}}), T(\phi_{Z_{j+11}})) + T(\phi_{Z_{m2}}) \ . \tag{3}$$

where $T(\phi_{Z_{ik}})$ ($i \leqslant m$, k=1 or 2) represents the total processing time of Z_i PCB's component set $\phi_{Z_{ik}}$ on machine k. Here, we assume components' processing time is independent from each other, which means $T(\phi_{Z_{ik}})$ can be calculated by the sum of the individual component processing time in $\phi_{Z_{ik}}$.

Since the makespan is highly connected with the particles' positions, we use equation (3) as the fitness function $f(x)$ to evaluate their obtained position from iteration to iteration and thus distinguish those better particles from those worse ones during the updating process. Hence, the updating mechanism can be expressed more specifically as follows: in the particles' level, if $f(x_{i,j}(t)) \geq f(pb_{i,j}^{t-1})$ ($pb_{i,j}^{t-1}$ is the best position of particle i on dimension j found so far at iteration t-1), then $pb_{i,j}^{t} = pb_{i,j}^{t-1}$, which means

the particle for the time being is inferior to its former movement. In this case, there is no need to replace $pb_{i,j}^{t}$ with $x_{i,j}(t)$. On the contrary, if $f(x_{i,j}(t)) < f(pb_{i,j}^{t-1})$, then the makespan will be reduced by replacing $pb_{i,j}^{t}$ with $x_{i,j}(t)$. By the same token, when it comes to the global level, if $\min f(pb_{i,j}^{t}) < f(pb_{g,j}^{t-1})$, then the global best particle will be updated as $pb_{g,j}^{t} = pb_{i,j}^{t}$. Otherwise, $pb_{g,j}^{t} = pb_{g,j}^{t-1}$. The mathematical expression of this relationship can be expressed as follows:

$$pb_{i,j}^{t} = \begin{cases} x_{i,j}(t) & if \ \ f(x_{i,j}(t)) < f(pb_{i,j}^{t-1}) \\ pb_{i,j}^{t-1} & if \ \ f(x_{i,j}(t)) \geq f(pb_{i,j}^{t-1}) \end{cases} \quad (4) \quad (5)$$

$$pb_{g,j}^{t} = \begin{cases} pb_{i,j}^{t} & if \ \min f(pb_{i,j}^{t}) < f(pb_{g,j}^{t-1}) \\ pb_{g,j}^{t-1} & if \ \min f(pb_{i,j}^{t}) \geq f(pb_{g,j}^{t-1}) \end{cases} \quad (6) \quad (7)$$

3.4 Overall Procedure of PSO

Step 1: (*Initialization*) Initialize a *swarm_population* of particles with random position and velocity that uniformly distributed within the scope of [0, 5] in the search space, and equate iteration *t*=1.

Step 2: Sort and rank the particles according to SPV rule.

Step 3: (*Updating*) Repeat until *t* meets the maximal iteration *updating_iteration*.

Step 3.1: For each PCB particle, update its velocity and position.

Step 3.2: Sort and rank the PCB particles according to SPV rule.

Step 3.3: For each component particle, update its velocity and position according to the individual best and global best position provided by PCB particles.

Step 3.4: Sort and rank the component particles according to SPV rule.

Step 3.5: Calculate the fitness value based on the permutation of the PCB particles and corresponding component particles.

Step 3.6: Update $pb_{i,j}$ and $pb_{g,j}$ according to equations (4) (5) (6) (7).

Step 3.7: Iteration $t=t+1$.

Step 4: Output global best solution $pb_{g,j}$.

3.5 Construction of PSO&ILS

Since particles in PSO are likely to be trapped into local optima and there is little chance for PSO to provide a solution with high-precision, this article introduces ILS to help the global best particles jump out of their local optima and thus further explore a better solution. To put it in another way, the hybrid algorithm is a two-phase algorithm that adopts PSO as its first phase while implements ILS as the second phase to improve the solution yielded by PSO. In general, ILS focuses on the searching of subspace defined by the local optimal solution and makes perturbations to local minimum once it is trapped into local optima. The implementation of ILS depends on the local search neighborhood, kick strategy and backtracking mechanism. More details are shown as follows.

After PSO outputs its global best solution, local search is adopted in the form of swap and insert to further explore the neighborhood of the solution for a better one. And in this article, swap operation is applied to the components exclusively. Under the swap mechanism, for each PCB set $I(i)=\{\phi_{Z_i1}\ 0\ \phi_{Z_i2}\}$, one component is selected from the component set ϕ_{Z_i1} and another component is selected from the other set ϕ_{Z_i2} and we exchange their positions. Swap strategy will continue until the neighborhood has been exhausted and then we evaluate their fitness values according to equation (3) and decide whether to replace the original best solution with the best swapped one or not. This is the same for insert strategy. For the new global best solution set $I'(i) = \{\phi'_{Z_i1}\ 0\ \phi'_{Z_i2}\}$ (there is chance that $I'(i) = I(i)$), choose one component from component set ϕ'_{Z_i1} (ϕ'_{Z_i2}) and insert it into the set of ϕ'_{Z_i2} (ϕ'_{Z_i1}) and this operation will carry on until the same termination condition is met as in the swap operation.

When the implementation of swap and insert has exhausted all the possible neighborhoods of the global best solution, the solution will be trapped into local optima again. To strengthen the algorithm's searching ability, ensuing kick strategy is adopted (in this article, kick strategy is applied to PCB exclusively). After local search outputs its local best solution, a temporary counter is generated to record the times that the best solution has not been updated. Every time the solution is trapped into local optima, kick strength is determined according to the temporary counter on an increasing scale and kick strategy will make perturbation to the PCB's permutation set based on the global best solution. For example, if the global best PCB permutation set is $Z=\{Z_1\ Z_2 \ldots Z_m\}$ and the temporary counter is b, then the kicked permutation is formed by rotating Z on the axis of PCB Z_b, that is $Z'=\{Z_{b+1}\ Z_{b+2} \ldots Z_m\ Z_1\ Z_2 \ldots Z_b\}$. If after the local search of Z', the global best solution can not be improved either, Z' will backtrack to Z and this time the rotation will base on the PCB Z_{b+1}. Since during this change the local optimal permutation of the components stays unaltered, the information of the current local best solution can be partly inherited by the new solution. Thus, there are more possibilities for the solution to jump out of the local optima. Once we find a better solution, the temporary counter will be returned to 0. This kind of operation will carry on until the maximal iteration is met.

4 Computational Result

The data regarding the mounting time of different components on different machines are generated with reference to the real-world production. And we use Visual C++ 6.0 to code our proposed PSO&ILS and other comparator algorithms to evaluate the effectiveness of this hybrid algorithm.

4.1 Parameter Setting

Since parameters w, r_1, r_2 affect the performance of PSO to a large extent, it is necessary to carry out experiments to confirm a relatively better combination of the parameters to help enhance the performance of PSO. In the literature, the suggested

scope of the inertia weight w is [0, 1.4], r_1 and r_2 belong to the scope of [0, 1] and under most circumstances, c_1=c_2=2.0.

Through numerous experiments and tuning, a relatively ideal configuration of the parameters should be: w=1.0, r_1=0.9, r_2=1.0, and *swarm_population*=25, *updating_iteration*=50. Also, according to literature [6], PSO can result in a better solution by means of eliminating the predetermined activity constraints on the particles' movement. Thus, we define a relatively large scale of activity scope for the particles, that is $x_{max} = v_{max} = 10^{49}$ while $x_{min} = v_{min} = -10^{49}$.

As to the maximal iteration of ILS operation, we set t=6 since under this circumstance PSO&ILS can result in a satisfactory solution while at the same time not taking up too much computation time under experimental conditions.

4.2 Comparison between PSO&ILS and PSO, PSO&ILS and Multi-Start Descent Algorithm

To illustrate the superiority of the introduction of ILS into the framework of PSO, a comparison between PSO&ILS and single PSO is made. Also, to verify the effectiveness of PSO&ILS in solving FSBPTA, multi-start descent algorithm (MSDA) with similar search scope as the proposed PSO&ILS is advanced for comparison. Several experiments carried out on an increasing scale of both PCB numbers and component numbers are shown in table 2, where "Dimension m×n" column denotes the scale of the tasks, and m represents the amount of PCBs and n represents the amount of components on each PCB. For example, 3×5 means there are altogether 3 PCBs and each PCB has 5 components. This is the same for the rest of the data in the dimension column. "Obj." represents the final makespan output of FSBPTA, "PSO&ILS vs PSO" denotes the improvement of PSO&ILS as compared with single PSO. "PSO&ILS vs MSDA" denotes the improvement of PSO&ILS as compared with MSDA. The CPU time is listed only for reference since it is affected by many factors, such as the settings of the compiler, the configuration of the computer, etc.

From table 2, we can note that PSO&ILS has achieved a 9.80% improvement as compared with PSO, which substantiates our claims about the effectiveness of ILS strategy applied to PSO. The average improvement of PSO&ILS as compared with MSDA in solving FSBPTA is approximately 3.70%, which proves the effectiveness of the proposed hybrid algorithm.

Also, from the data in table 2, we can know that when the dimension is relatively small, MSDA may attain a better solution than PSO&ILS within the same CPU time. However, as the dimension increases, the superiority of PSO&ILS over MSDA is quite evident. Thus, for the real-world application, PSO&ILS has its undeniable advantage.

Nevertheless, we should also be aware of the price of this improvement. As the dimension increases, while the CPU time of PSO&ILS and that of MSDA stay within the same scale, the CPU time of PSO&ILS and that of PSO differ substantially. Thus, we must call for a balance between solution quality and computation time in reality, which can be achieved by adjusting the maximal iteration of ILS.

Table 2. Comparison between PSO&ILS and PSO, PSO&ILS and MSDA

Dimension	PSO		MSDA		PSO&ILS		Improvement (%)	
$m \times n$	Obj.	CPU time(s)	Obj.	CPU time(s)	Obj.	CPU time(s)	PSO&ILS vs PSO	PSO&ILS vs MSDA
3×5	11.70	0.25	12.00	0.28	11.70	0.27	0.00	2.50
3×10	25.90	0.39	25.70	0.45	25.60	0.44	1.16	0.39
3×20	64.60	0.61	67.90	0.89	62.30	0.88	3.56	8.25
3×50	180.20	1.45	178.40	3.39	167.40	3.41	7.10	6.17
4×5	22.20	0.34	23.30	0.42	22.20	0.42	0.00	4.72
4×10	38.80	0.53	36.60	0.59	36.50	0.58	5.93	0.27
4×20	84.50	0.88	81.90	1.23	78.80	1.19	6.75	3.79
4×50	220.20	1.94	204.70	4.72	196.00	4.44	10.99	4.25
5×5	27.90	0.47	27.60	0.50	27.20	0.50	2.51	1.45
5×10	43.00	0.69	42.70	0.84	40.60	0.83	5.58	4.92
5×20	105.50	1.16	102.70	1.77	97.50	1.72	7.58	5.06
5×50	284.70	2.47	278.10	9.98	259.40	9.44	8.89	6.72
8×5	45.80	0.83	45.80	0.89	44.10	0.88	3.71	3.71
8×10	80.00	1.27	76.20	1.55	75.40	1.52	5.75	1.05
8×20	168.90	1.91	149.80	3.42	150.20	3.41	11.07	-0.27
8×50	445.30	4.02	427.40	24.24	386.60	22.23	13.18	9.55
10×5	60.20	1.19	55.00	1.36	55.80	1.34	7.31	-1.45
10×10	106.50	1.59	102.90	2.05	96.40	2.00	9.48	6.32
10×20	224.70	2.53	206.10	4.89	202.60	4.58	9.84	1.70
10×50	576.40	5.67	558.40	34.67	507.90	33.55	11.88	9.04
15×5	93.60	2.23	86.50	2.53	87.40	2.52	6.62	-1.04
15×10	168.80	3.00	154.60	4.31	154.40	4.22	8.53	0.13
15×20	355.70	4.78	321.30	11.45	302.50	10.83	14.96	5.85
15×50	897.00	10.09	792.60	78.56	741.80	76.73	17.30	6.41
20×5	134.30	3.53	117.20	4.06	117.50	4.02	12.51	-0.26
20×10	244.50	4.16	211.70	6.45	210.10	6.27	14.07	0.76
20×20	478.50	6.95	432.10	17.36	404.00	16.75	15.57	6.50
20×50	1208.60	14.75	1072.00	129.69	998.10	121.53	17.42	6.89
50×5	349.60	16.64	292.80	20.92	285.60	20.80	18.31	2.46
50×10	628.10	21.52	524.80	37.16	506.30	36.72	19.39	3.53
50×20	1241.70	27.80	1077.80	111.75	1023.80	103.19	17.55	5.01
50×50	3166.60	52.02	2673.80	889.74	2564.40	805.13	19.02	4.09
Average improvement (%)							9.80	3.70

5 Conclusion

From the experiments, we can note that the introduction of ILS into PSO has improved the solution quality substantially. According to the computational results, it is evident that the proposed PSO&ILS is statistically and significantly better than either PSO or multi-start descent algorithm, especially when the total amount of tasks is a large one.

With this encouraging and promising application of PSO in the flowshop-based parallel task assignment problem, we are sure that ensuing efforts put on the research

of PSO and continuous work to pursue better performance of PSO is well-deserved. Especially when we take account of the imperfect characteristics inherent within the frame of PSO, further research on the introduction of other effective and efficient strategies from some successful algorithms will be helpful in enhancing the performance of PSO.

Acknowledgments. We thank anonymous referees for their constructive comments. This research is partly supported by National Natural Science Youth Foundation for Distinguished Young Scholars of China (Grant No. 60804053).

References

1. Lenstra, J.K., Rinnooy, K.A., Brucker, P.: Complexity of machine scheduling problems. Annals of Discrete Mathematics 1, 343–405 (1977)
2. Ali, A., Fawaz, S., Al-Anzi: A PSO and a Tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. Computers & Operations Research 33, 1056–1080 (2006)
3. Zhang, C., et al.: An improved particle swarm optimization algorithm for flowshop scheduling problem. Information Processing Letters 1016, 5–15 (2008)
4. Kashan, A.H., Karimi, B.: A discrete particle swarm optimization algorithm for scheduling parallel machines. Computers & Industrial Engineering (in press, 2008)
5. Tasgetiren, M.F., Sevkli, M., Liang, Y.C., Gencyilmaz, G.: Particle swarm optimization algorithm for permutation flowshop sequencing problem. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 382–389. Springer, Heidelberg (2004)
6. Liu, Z., Wang, S.: Research on parallel scheduling problem based on particle swarm optimization algorithm. Computer Integrated Manufacturing Systems 12, 183–188 (2006)

Target Localization for Autonomous Soccer Robot Based on Vision Perception

Guifang Shao[1], Yuhua Wen[2,*], Fang Yu[3], and Zushu Li[3]

[1] Institute of Pattern Recognition and Intelligent System, Xiamen University, Xiamen 361005, P.R. China
gfshao@xmu.edu.cn
[2] Department of Physics, Xiamen University, Xiamen 361005, P.R. China
yhwen@xmu.edu.cn
[3] Intelligent Automation Institute, Chongqing University, Chongqing 400030, P.R. China
fang.yu@intel.com, zushuli@vip.sina.com

Abstract. In order to solve the target localization question for autonomous soccer robot, a coordinate space transformation model was developed. Based on the characteristic analysis of camera, we use panoramic vision to adjust robot posture, use foreground vision to count the exact distance between object and robot, and calculate the vertically distance and horizontal distance through transform frame coordinate system to world coordinate system. Contrastive experiment results prove that this algorithm is accurate and reliable, it also be successfully used for object reorganization and position judgment at RoboCup competition.

1 Introduction

Most of the soccer robot systems depend on sensing system to make decision, while sensing system usually takes camera image processing as main means, so how to locate target based on vision perception is very important. In order to locate target accurately, there are two important things must be done, one is to make sure the object real position under robot world coordinate system, the other is to fusion the sensing information gained by multi-sense devices. Currently, there are many monocular vision distance detection algorithms which had been applied widely and successfully[1,2], and there are many literatures talked about the object depth coordinate computing methods[3-6].

But in real complex environment of soccer robot competition, target may be stand on every direction towards robot, not only just the front, so, merely acquire the vertical distance between target and robot is not enough. We should get the exact position coordinate, i.e., not only includes the vertical distance but also the horizontal distance. Literature [7] introduced a method to compute the object lateral coordinate which only need camera inside parameters, camera vertical height and declining angle. While literature [8] taken two more parameters to count it besides those in literature [7], one is the closet distance of camera vertical viewing angle projection to ground, another is the

* Corresponding author.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 831–840, 2008.
© Springer-Verlag Berlin Heidelberg 2008

farthest distance of camera horizontal viewing angle projection to ground. These methods can limit error to a narrow range within 1miles, when the detection distance added to 1.2miles, however, it grown to 20cm, and the bigger the distance is, the greater the error will be, thus limited its application.

In order to provide a more accurate and more reliable position information, and locate target accurately within more widely detecting distance, this paper puts forward a new algorithm which combines the information come from foreground vision and panoramic vision, at the same time, it computes the vertical distance and the horizontal distance between robot and target. This paper took Frontier-I as experiment object which is an autonomous robot and designed by Shanghai Jiao Tong university.

2 Vision Characteristic Analysis

The Frontier-I robot has two cameras for vision perception, one is placed in front and one is placed at above. During to the difference of installation place and the camera feature itself, in actual competition, they should play different roles, but how to assign the role and how to compute some useful information from it? In order to solve these questions, first we must know about the actual performance of camera, and then we can utilize its advantages properly.

2.1 Foreground Vision Analyzing

During to the camera physical character and the effects come from different installation means, there are some characteristic differences between the images taken from far and near. Robot foreground view range looks like a rectangle, as shown in figure 1(a). In order to learn more about the camera vision character, according to figure 1(b), we set some sampling points in actual competition field. During the first 3 miles of view range, we place the ball on every sampling point and record its coordinate values counted from the visual image. Then, according to the coordinate x, y of each sampling point we plot figure 2 by Matlab 6.5.

Contrast figure 1(b) with figure 1(c), we can know that those well-proportioned sampling points in actual game field will be non-uniform after collection by camera, and become nonlinear enhancement from near to far. Experiment results indicate that the linearity is very good within 2 miles, but with the distance increased the nonlinearity also increased. Base on this feature, we can conclude that when target is far away from robot, its location information is vague, the nearer to the robot, the more accurate the positioning information will be, so we can do localization accurately with foreground vision, such as the exact distance of object .

2.2 Panoramic Vision Analyzing

The panoramic vision system consists of camera, image collection card, cone mirror and sustain machine. For the mirror reflecting and lens aberration, panoramic camera has higher nonlinearity. Especially because the top of cone mirror is so tine that the object mapped on it becomes very small, thus it can't be recognized after reflected into

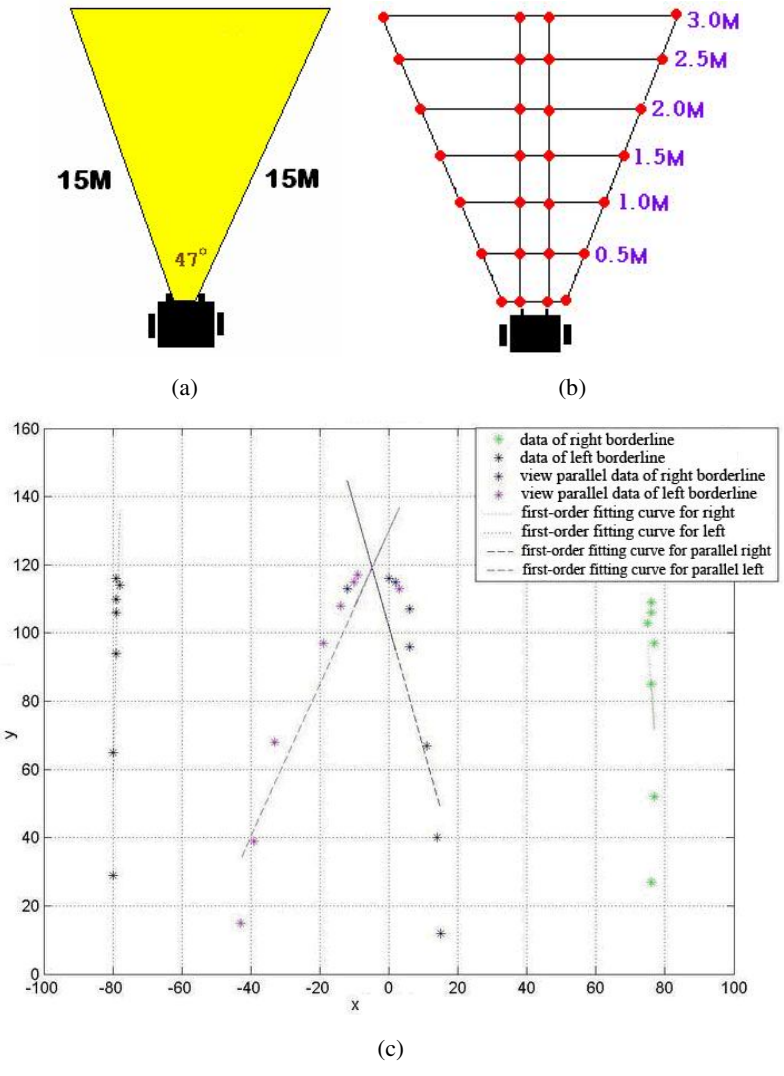


Fig. 1. Foreground vision character. (a) Foreground visual range. (b)Test setting. (c) Test results of foreground actual character.

an image. Therefore, a vision dead zone comes into being as shown in figure 2(a), in which may be the object is too small (for example, the ball) to be recognized, or may be it is big enough, for instance, the gate, gate pillar or other robot, due to its height that its top can be reflected into image far away from top of cone.

Through the actual experiments we can know that there is a vision dead zone within 50cm. In order to do panoramic vision research deeply, we design a vision perception testing experiment as follows:

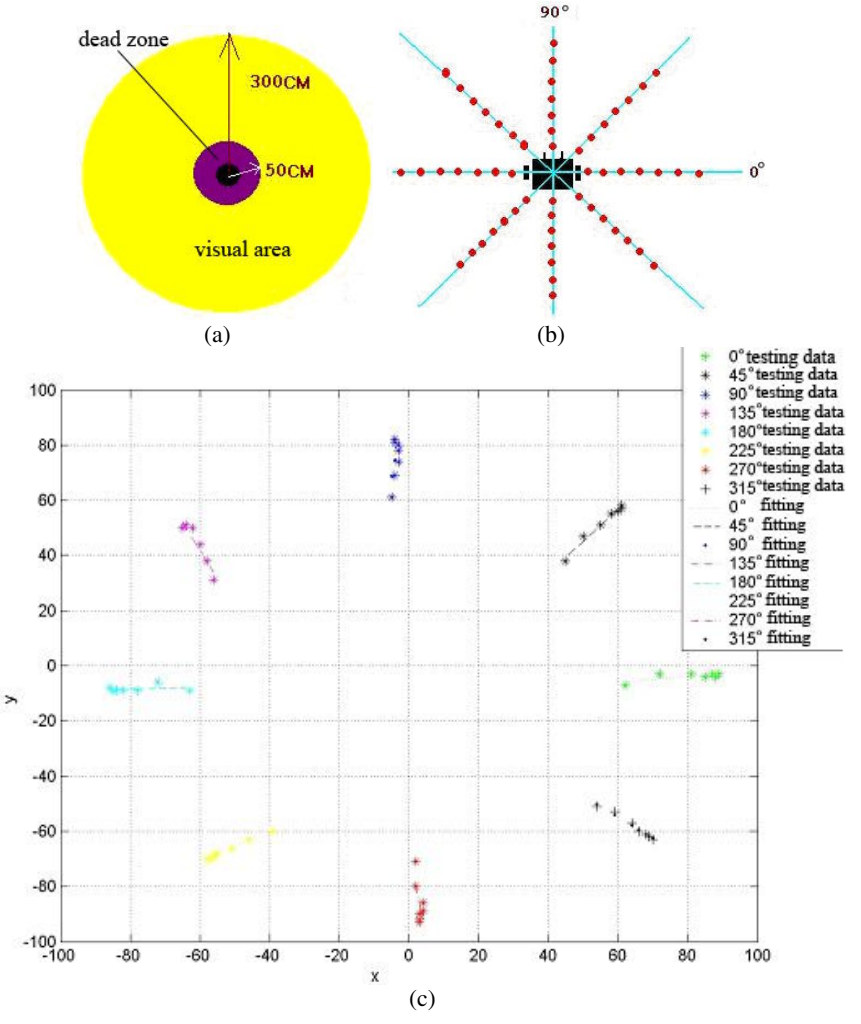


Fig. 2. Panoramic vision character.(a) Panoramic visual range. (b) Test setting. (c) Test results of panoramic actual character.

Taking robot positive direction as 90 degree, horizontal right direction as 0 degree, robot body center as origin, we create a new coordinate. Making goal pillar as the vision testing target, in 0 degree direction, tests goal pillar center coordinate (x, y) at different distance, from 50cm to 350cm, every 50cm tests once. Similarly, does the same tests in degree 45, 90,135,180,225,270 and 315, as shown in figure 2(b).

Using matlab 6.5 plot all the x, y coordinate values, we can get the result as figure 2(c) shows, its nonlinearity is higher and image distort greatly, especially, when beyond 200cm, near or far of distance in image can't be recognized, but we can use it to count angle, so the panoramic vision is suitable for probable location.

3 Algorithm Realization

According to the analysis in 2, foreground vision non linearity is smaller than panoramic vision and has higher reliability, so we can use it to does accurate action control, while the panoramic vision can be used to do probable localization. Based on the actual characters of foreground vision and panoramic vision, we proposed that using panoramic vision to do zoning localization when target is far away, i.e., computing the moving angle for robot, then making use of foreground vision to count the vertical distance Y and horizontal distance X while target near to robot.

3.1 Lateral Localization in Horizontal Direction

As shown in figure 3, there is a declining lens, o is its optical center, f is the focus, c is the optical axis, h is the distance of o to ground, o' is the intersection point of optical axis with image plane, α is the included angle between c and horizontal direction, γ is the include angle between c and oo', β is the included angle between oo' and ground. (f_u, f_v, u_0, v_0 are camera inside parameters).

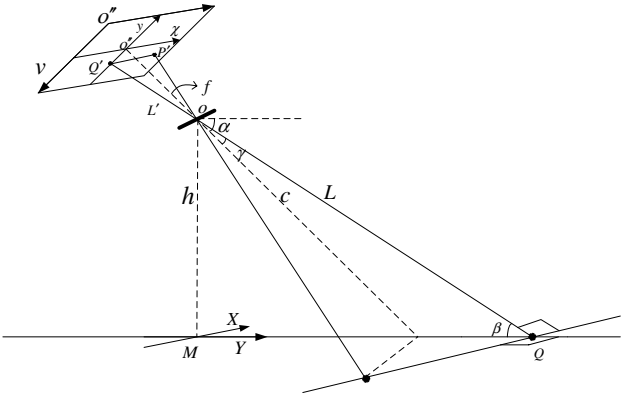


Fig. 3. Horizontal imaging

In order to get the accurate distance between object and the robot, we define 3 coordinates as followings:

(1)World coordinate system XMY , Which taking camera projection M as origin, taking the left direction of camera parallel to horizontal line as X positive direction, taking the fore direction of ground perpendicular to camera as Y positive direction.

(2)Image plane coordinate system $xo'y$,which taking the image plane center as origin and the length as unit.

(3)Frame storing image plane coordinate system $uo''v$,which taking image plane top&left angle as origin and the unit is pixel.

Mapping two points P, Q in image plane $xo'y$ and the corresponding image point are P', Q' . Under the world coordinate system XMY , let $P(X, Y), Q(0, Y)$, while in the frame

storing image plane coordinate system $uo''v$, let $P'(u,v), Q'(u,v)$, $o'(u_0, v_0)$, s_u, s_v are the length and width of a pixel, then there is:

$$\beta = \alpha - \gamma, \gamma = \alpha \tan(y / f), l = \frac{h}{\sin \beta} = \frac{h}{\sin(\alpha - \alpha \tan(y / f))} \tag{1}$$

Changing equation (1) into pixel coordinate, such as

$$\begin{gathered} l = \frac{h}{\sin(\alpha - \alpha \tan \frac{(v - v_0)s_v}{f})} = \frac{h}{\sin(\alpha + \alpha \tan \frac{v_0 - v}{f})} \\ l' = \sqrt{f^2 + y^2} = \sqrt{f^2 + (v - v_0)^2 {s_v}^2} \end{gathered} \tag{2}$$

Because $X = xl / l'$ and $x = (u_0 - u) \cdot s_u$, so

$$X = \frac{(u_0 - u)s_u \cdot h}{\sqrt{f^2 + (v_0 - v)^2 {s_v}^2} \sin(\alpha + \alpha \tan \frac{v_0 - v}{f_v})} \tag{3}$$

3.2 Depth Localization in Vertical Direction

In order to make sure the exact distance between object and the robot, after computing the vertical distance X , it also needs to calculate the horizontal distance Y .

As shown in figure 4, through lens the point P can be mapped in o' and produce image point P' .

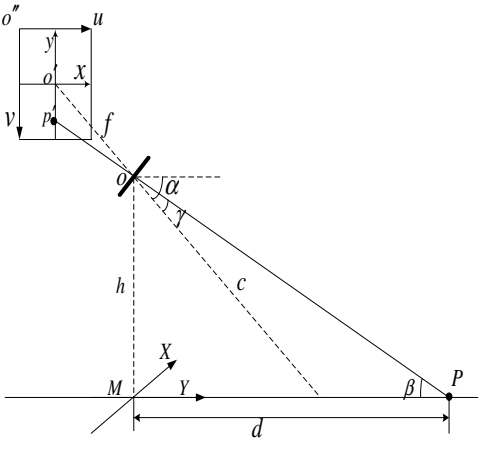


Fig. 4. Vertical imaging

Under $uo''v$, let coordinate o' is (u_0, v_0), at the same time, let coordinate P' is ($0, y$) under image plane coordinate system $xo'y$, such that

$$\beta = \alpha - \gamma, \gamma = \tan(y / f), Y = h / \tan(\beta) \tag{4}$$

Because the value counted from image doesn't represent the real y, but the pixel numbers v, and they have the relation as $y = (v_0 - v) \times s_v$, where s_v is a pixel length, so we can get Y from formula (5) as follows

$$Y = \frac{h}{\tan(\alpha + a\tan(\frac{v_0 - v}{f_v}))} \qquad f_v = \frac{f}{s_v} \tag{5}$$

Therefore, we get a set of coordinate conversion formulas from frame storing image plane coordinate system $uo''v$ to world coordinate system XMY, and realize the mapping from image plane T' to actual ground plane T.

3.3 Zoning Localization

It is unnecessary to locating the target accurately when it far away from robot, at this time we only want to know whether it should be traced and make sure the moving direction towards it. For panoramic camera can see the target in farther distance, and the soccer robot game and its field have special rules, for example, the opposite goal is very clear and when robot moving into some zone it should attacking or defending, so this paper puts forward that separate the game field into various little areas according to different function, as shown in figure 5.

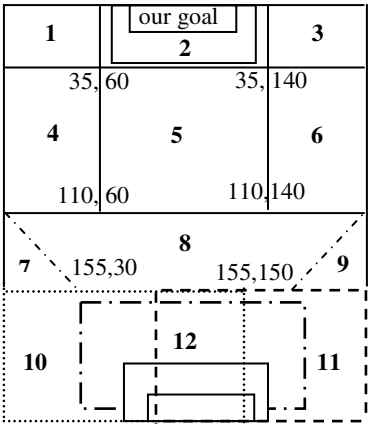


Fig. 5. Sketch map of field zoning

Each zone represents different function, for instance, zone 1,2,3 are the defending area, while zone 10,11,12 are the attacking area. The zoning localization method can play many roles, one is to locate target and return θ, d, i.e., angle and distance, another is to guide robot to make decision and select different strategy.

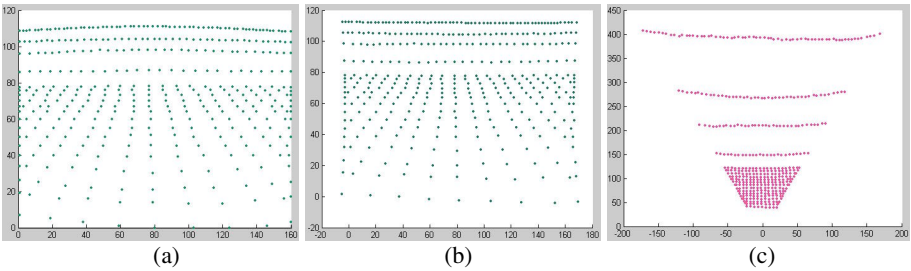
4 Experiment and Analysis

4.1 Experiment

In order to verify the algorithm proposed in this article, we place a grid cloth on the game field, each grid size is 6*6cm, taking each gird vertex as sampling point, recording its frame storing coordinate($u(i), v(i)$), and does the aspheric correction for pixel coordinate of sampling point. Then, through formula (3),(5) mapping it to the world coordinate system, gain the computing coordinate($X(i), Y(i)$).The camera parameters are $f_v = 200, f_u = 200, u_0 = 80, v_0 = 60$.

Figure 6(a) shows the $u(i), v(i)$ distribution of sampling points, which taking pixel as unit. As to camera optical axis doesn't vertical to ground and has a little declining angle, though the grids on cloth are well-proportioned, but the frame storing coordinates are inhomogeneous from near to far in X, Y direction. Figure 6(b) is the adjustment image based on aspheric correction method.

Put $u(i), v(i)$ into formula (3) and (5), we can get a set of coordinates $(X(i), Y(i))$, and plot them under world coordinate system XMY as figure 6(c) shown.



(a) (b) (c)

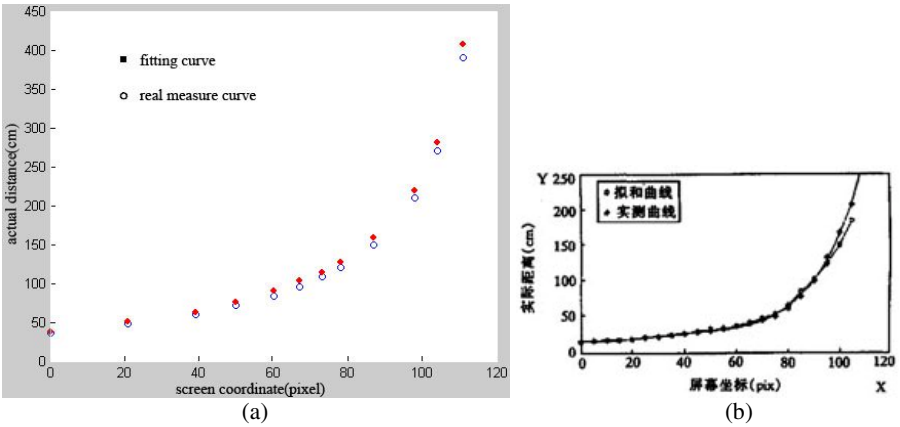
Fig. 6. Experiment setup. (a) Sampling points distribution on image. (b) Sampling points distribution after emendation. (c) Calculate values of sampling points.

4.2 Results Analyzing

Table 1 gives some different data of actual distance, computing distance and their error, where AV means Actual Value, CV represents Computing Value, AE denotes Absolute Error, i.e., $AE = |CV - AV|$, RE is Relative Error.

Table 1. Calculated value and error value

name	X					Y					
AV cm	6	18	30	48	54	36	60	84	108	210	390
CV cm	6.7	19.9	32	51.9	57.7	37.9	63	90	114.3	218.7	407.2
AE cm	0.7	1.9	2	3.9	3.7	1.9	3	6	6.3	8.7	17.2
RE %	1.17	3.17	3.33	6.5	6.17	0.3	0.75	1.5	1.58	2.18	4.3



(a) (b)

Fig. 7. Difference between calculate value and measure value

According to the experiment results we can know that the algorithm proposed in this paper can realize the conversion from image plane coordinate system to actual world coordinate system.

Figure 7(a) gives the difference between calculate value and measure value in this paper and Figure 7(b) shows that in literature [7]. From the result we can know that method in literature [7] can realize target localization and get good results within 100cm, when distance added to 100cm-120cm the error increased to 20cm. When distance reaches to 120cm the calculated result in this paper is smaller than 8cm, it is less than 10cm while distance is 150cm, and it littler than 20cm when distance is 390cm. The method proposed in this paper can obviously restrict error to a smaller range.

5 Conclusions

Based on the different characteristic of foreground vision and panoramic vision, this paper puts forward that using panoramic vision to compute the moving angle when target is far away from robot, and making use of foreground vision to calculate the vertical and horizontal distance while target near to robot. This paper introduces how to gain the target coordinate in 2-D space, and does a lot of experiments to verify the feasibility and validity of this algorithm.

Acknowledgements. This paper is supported by the National Natural Science Foundation of China (Grant No. 60443004, 60274022), and the Natural Science Foundation Project of CQ CSTC (Grant No. 2006BB2395 and Grant No.2007BB2415). Gratefully acknowledges Professor Zushu Li for his guiding work in this paper.

References

1. Shen, Z., Huang, X.: Monocular vision distance detection algorithm based on data regression modeling. Computer Engineering and Applications 43(24), 15–18 (2007)
2. Zeng, S.: Research of distance detection algorithms based on monocular vision. Kunming university of science and technology, master degree paper, 1–18 (2007)

3. Xie, Y., Yang, Y.: A self-localization method with monocular vision for autonomous soccer robot. Microelectronics & Computer 22(10), 129–132 (2005)
4. Yu, F., Li, Z., Wang, N.: New method for autonomous soccer monocular vision position determination. Computer Measurement &Control 15(27), 1781–1784 (2007)
5. Li, M., Hong, B.: Monocular-vision-based mobile robot global localization. Robot. 29(2), 140–144 (2007)
6. Chen, G., Sun, X., et al.: Monitor system of independent motion robot based on monocular visual. Development & Innovation of Machinery & Electrical Products 20(1), 4–6 (2007)
7. Jiang, Y., Gao, Q., et al.: Research on Recognition and Localization Methods for Autonomous Soccer Robot. Journal of Northeast China Institute of Electric Power Engineering 22(3), 12–16 (2002)
8. Zhu, Y., Hong, B., Ruan, Y.: Fast recognition and localization of objects by full-autonomous soccer robot. Journal of Harbin Institute of Technology 35(9) (2003)

Urban Land Classification Research Based on Data Field

Yu Deng[1,2], Haijun Wang[3,*], Li Wang [2,4], and Dandan Shan[3]

[1] Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing 100101
[2] Graduate University of Chinese Academy of Sciences, Beijing 100049
[3] School of Resources and Environment Science, Wuhan University, Wuhan 430079
landgiswhj@163.com
[4] Institute of Policy and Management, Chinese Academy of Sciences Beijing 100080

Abstract. This paper further studied the theory of urban land classification based on data field starting from the practice of the urban Land classification work, relying on the basic principles of data field and proposed "the urban land classification methods based on data field". It's basic idea is: first, determining the type of the classification factors based on the way and space features of their impact on land. Subsequently, determining the approach of anisotropy and overlay of the radius of influence and the range of radiation according to the inherent characteristics of different classification factors. Finally, calculating different types of factors in different calculation method to obtain a total score of these classification units under specific form of role. In this paper pointed out the implementation steps of this method, and successfully proved that the method is practical and scientific by giving an example.

1 Introduction

Urban land classification plays a key role in the configuration of land resources and optimization of land use structure. As the continuous improvement of market mechanisms in China, urban construction and planning has a more close relationship with the basic work which is "self-examination and location". Urban Land classification has an important guiding significance in advancing all of the causes of the city and the relevant institutional arrangement. At present, some methods of China's urban land classification are: the methods of multi-factor comprehensive appraisal, the methods of differential profit, gradation by the land price, and so on[1-4]. The general approach of Chinese urban land classification is: using multi-factor comprehensive assessment to determine the urban land-level, then, using the methods of differential profit and other methods to determine the final land-level. The major steps of multi-factor comprehensive assessment is: choosing the factors and calculating the weight, dividing the unit of classification, calculating the classification factors at function and service radius, calculating the role of sub-classification unit, calculating the total score of these units, land-delineation of land-level examination.

* Corresponding author. National Natural Science Foundation of China sustentation item (40871179).

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 841–848, 2008.
© Springer-Verlag Berlin Heidelberg 2008

This paper proposed the method of urban land classification based on data field, for objective, and this method can be used to estimate the sub-classification unit accurately. Data field reflects the General contact and interaction mechanism between the space objects. The theory of data field pointed out that space object gathers and disperses constantly, and then demonstrates the existence of itself through radiation potential energy[5]. The change of potential energy meets the law of friction, and it decay in a specific way to describe the impact of space objects. This method can describe the potential distribution of space, measure the potential nature of space and the value of related classification units. At the same time, it gives a way of potential attenuation, and provides a perspective on the linkage between radius and potential energy based on the impact of factors.

2 Extended Research on Data Field

2.1 Data Field

In physics, the physical quantity located in a certain region often referred to as "field", such as the gravitational field and the electric field. In the data domain space, each data radiates energy to the entire space, to show their existence in the task of data Mining and Knowledge Discovery, so there is the data field. If the radiation energy of the data is directional, the data field is a vector field, Otherwise, it is a scalar field. Field data has the general features just like all the fields, but it is not entirely the same as the field model. The existence of data field, must meet some conditions such as independence, the nearest of the traverse, stacking, and the attenuation of isotropic, and so on[6-7].

Data Field originated in radiation data, it is a space description of the radiation data. And it is a mathematical abstraction and assumptions of the radiatical process . The introduction of market solves this problem well. The scope of radiation based on different points can be expressed more accurately with potential and attenuation function. The potential function of point x in the data Field is defined as the total affect from all the data points. If there are n data points, the distance D = (d1,..., Dn), then the potential function on x is defined as:

$$F(x) = \sum_{i=1}^{n} \rho_i e^{-\frac{(d(x,di))^2}{2\delta^2}}$$

Potential function is a function of distance or location, it can be superimposed, so each data object in the data domain space makes a contribution to the potential of all the points in the field , and the contributions are inversely proportional to the square of distance. d is the the distance from x to data point d_i, ρ_i is the value of the point, σ is the radiation factor. We can get d (x , di) based on ρi, then get the potential of all the points ,and the superposition of the potential of all the points gets the total value of potential F.

It is worth mentioning that the change of radiation factor σ will directly impact the space of the equipotential line, if σ is small, the impact of a single data point is small,

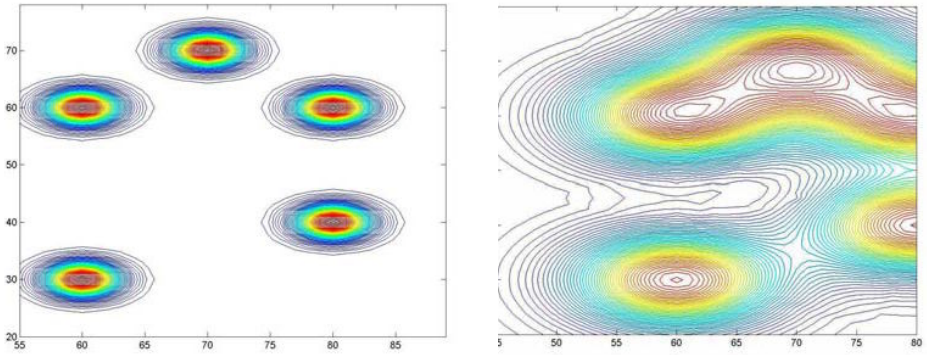


Fig. 1. 2D data field with different σ （σ=2、σ=6）

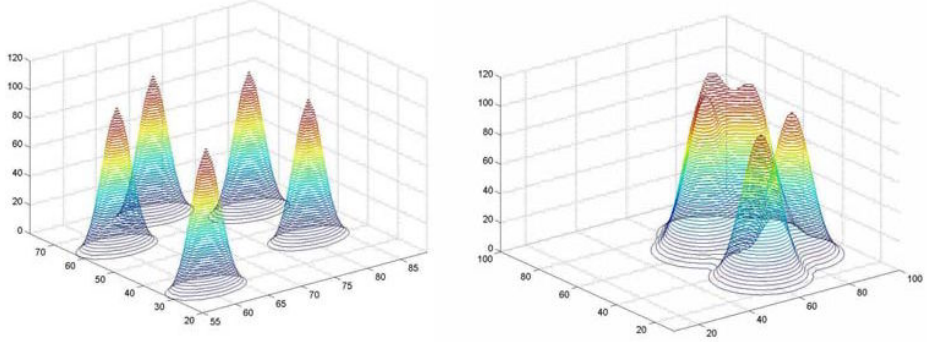


Fig. 2. 3D data field with different σ （σ=2、σ=6）

and so is the line of equipotential; if σ is big, the impact of a single data point is big, and so is the line of equipotential. As shown in Figure 1, 2:

When the distance is zero, the potential function of a single object is the largest(=1); when the distance is infinity, the potential is the minimum (to zero). In fact, according to the "3σ rules" of normal distribution, when the distance is larger than or equal 3σ, the the value of the potential function is approximate zero, therefore, the radius of influence of each object in the field can be approximated to: r≈3σ.

2.2 Prolate Theory of Data Field

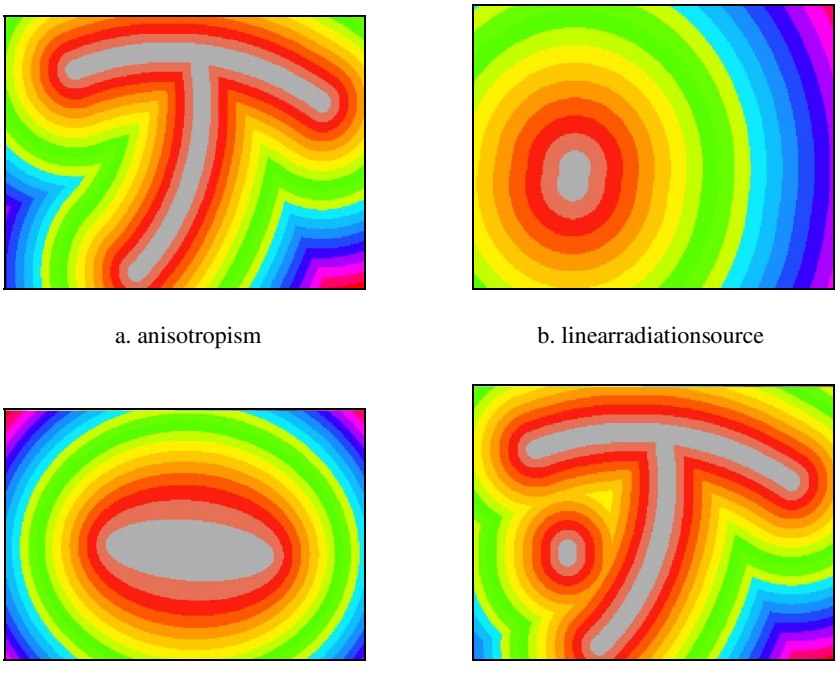
Data field impacts the space objects in the form of radiation, as well as the interaction between the them. The quantitative expression is based on potential, potential energy gradually decline from center to the edge. According to this theory, "potential flow" of the objects in different size and different levels in the space spreads from inside to

outside, from high to low in the way of potential, thus we can get an accurate measurement of the spatial distribution. However, the traditional data field is a radiation field based on the point spatial object, although point element is the basic component of space objects, but this single expression and the way of factor reconstruct appears stretched in a simulated form of classification factors.

Because of such considerations, in order to meet the needs of urban land classification, this paper puts forward the prolate theory of data field, the theory is a complement and perfection of data field, mainly in the following four aspects:

2.2.1 Radiant Anisotropy of the Classification Factors

In the complicated world, homogeneous and symmetry is very difficult to meet. For example, definition of tihe nfluence radius of the service center is usually unsymmetrical, it may spread in a certain direction along the Traffic Trunk Road, or spread towards the residential areas. As shown in Figure 3-a.



a. anisotropism b. linearradiationsource

c. radiation of combined factor d. overlay operation （vector sum）

Fig. 3. Schematic diagram of extended data field

2.2.2 Radiation of Linear Factors

Linear factors like linear distribution facilities impact the classification unit along the linear facilities, and the impact attenuates from near to far. As shown in Figure 3-b.

2.2.3 Resultant Radiation of Polymorphic Classification Factors

The real world is a combination of various factors, such as point, line and polygon, and they are always complex and irregular, therefore, the expression of polymorphic factors is particularly important. As shown in Figure 3-c.

2.2.4 Algorithm on Overlaying

The value of the classfication unit is the weighted average value of all the classfication factors. And this relates to the overlaying of factors distribution and its radiant range. Only taking the vector sum into consideration is not strict, such as the comprehensive study of the impact of three adjacent businese centre to its service centres point, we can measure it by the way of sum, max, average, and so on. As shown in Figure 3-d.

3 Urban Land Classification Research Based on Data Field

The basic idea of urban land classification based on data field is: first of all, definting the type of the fators according to the the way how do the factors impact on land and their space characteristic; Subsequently, according to the inherent characteristic of different classification factors to determine the impact radius, the anisotropy of radiant range and the overlaying manner; Finally, calculating in accordance with different calculation types of functional score, to obtain the total score of the classification units in specific forms of impact.

3.1 Divide of the Classification Factors

Divide the type of the factors into punctate factors, linear factors and polygon factors according to the the way how the factors impact on the classification units and their space characteristic. Punctate factors are the factors which distributed in the urban region like points, their impact on classification units are relative to the relevant distance from facilities such as theaters, schools, business service centre. Linear factors are factors which distribute along facilities, and their impact on classification units attenuates from near to far with its centre of linear factors facilities. Polyon factors distribute evenly like flakes, they cover a integral region, and their impact on classification units is only relevant to factors indicators, such as noise pollution, air pollution and so on[4].

3.2 Definition of the Impact Modes of ClassifIcation Factors

① Impact radius is closely related to the inherent attributes of classification factors, if the function is different, then the impact radius is different, take punctate factor (business service centre) as an example, as follows[8]:

Origin of the radius is the edge of a commercial service center, with the distribution of the commercial service center, we can get:

Service radius of a municipal centre=the maximum distance from the municipal centre to built-up areas

Service radius of other centres besides the municipal centre=the maximum service distance of the commercial service center which is at the same level

The relative distance was calculated according the following formula, service centres at different levels are divided into 3-15 regions:

$r = di \div d \quad [0 \le r \le 1]$

Where r is the relative distance; di is the distance from the centre to the point (di<the service radius); d is the service radius of the centre.

②The functional score of classification may be only relative to the relevant distance from the classification factors, and this is isotropic. In practical problems, anisotropy more common, and the functional score of with same relevant distance may be lager at a certain direction. This phenomenon is the direction effect of radiant range. And direction effect can characterize the space functional pattern of the classification factors more accurately.

③The interaction mode between space objects is complex, a objective, reasonable overlaying mode of functional score is conducive to describe the functional mechanism, to reflect the spatial pattern of functional score of classification factors accurately.

3.3 Calculation of the Functional Score of Classification Factors

Calculation mode of functional score of classification units is determined by the classification factors, although the three factors are not the same, the basic calculation mode is as follows:

①Punctate factors：

$$f_{ij} = \mathrm{F}_i \cdot e^{-\frac{(d(\mathrm{j},di))^2}{2\delta^2}}$$

Where f_{ij} is the functional score of the classification factor in classification unit; F_i is the functional score of the classification factor i ; d (j , di) is the distance from classification factor i to classification unit j; δ is the radiation factor ($r \approx 3\delta$).

②Linear factors: Calculation principle of the functional score of linear factors based on data field is the same as above.

③Polygon factors: Polyon factors distribute evenly like flakes, they cover a integral region, and their impact on classification units is only relevant to the factors indicator.

At last, we can obtain the total value of the classification unit using the weighted average value of all the classfication factors, finally, we get the grade of the urban land.

4 An Empirical Study

This study selects the urban land classification of Luoding City as research object and implements the grid points role computing method based on the data field. Luoding City locates in western Guangdong Province and the land classification area of which is 16.082 km^2 using 100m×100m fixed grid as a land classification unit. In accordance with the space characteristic of the classification factors, classifies them into points, lines and surface-like factors. Obtain the role value of classification unit according to the determine methods of each role factors. As shown in Figure 4 and Figure 5.In order to make the quantitative of the grid point role accurately, in actual operations, we can improve the measurement of potential energy from the following two aspects: ① non-symmetry radiation characteristics of the classification factors; ②the stack computing of potential energy can be taken in such computing means as for sum, maximum, minimum , average and so on.



Fig. 4. The Isoline Graph of Effect Value of points

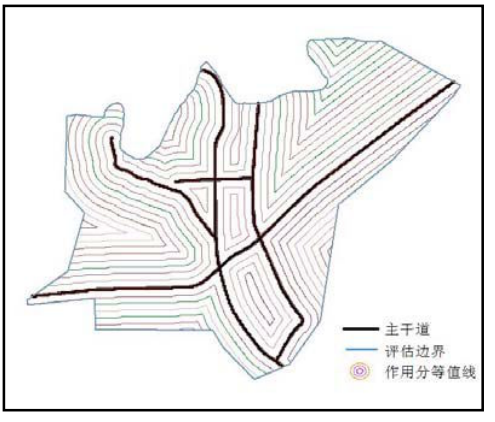


Fig. 5. The Isoline Graph of Effect Value of plotlines

5 Conclusion

Data field and its prolate theory effectively overcome the issue of polymorphic factors in urban land classification. And a satisfactory solution against the non-symmetry and multi-selective of overlaying algorithms of the factors role value is put forward so that urban land classification becomes more objective and reasonable. Through the land classification project of Luoding City, with true and reliable data validated this method; the experimental results are in line with the actual situation which further explained that urban land classification method based on data field is a scientific, strict and practical solution one.

References

1. Wang, H.j., Zhang, D.l.: Urban Land Gradation Method Based on Spatial Clustering. Geomatics and Information Science of Wuhan University 31(7), 629–633 (2006)
2. Yan, X., Lin, Z.: The Assessment of City Property. China Renmin University Press, Beijing (2000)
3. Yan, T.-l., Zhu, D.-h., Yang, Y.-x.: The model of land information system. China Land Science 10(4), 42–47 (1996)
4. Wang, H., Guan, X., Xia, W.: Urban Land Classification Method Considering Obstacle Space. Geomatics and Information Science of Wuhan University 33(1), 98–102 (2008)
5. Wang, S.: Data Field and Cloud Model Based Spatial Data Mining and Knowledge Discovery:[doctoral dissertation], Wuhan University (2002)
6. Dai, X.-j., Liu, C.-y., Han, X., Li, D.-y.: Application of data field in Information Token. Journal of Fudan University (Science Edition) 43(5) (2004)
7. Jin, W., Li, D.: Hierarchical Clustering Method Based on Data Fields. Acta Electronica Sinica 34(2) (2006)
8. General administration of quality superision, inspetion and quarantine of the people's republic of china. GB/T 1850722001. The regulations <app:regulations> of urban land classification. Standards of the Expansion Joints Manufacturers Association, Beijing (2001)

Research and Design of Digital Synthesizer Based on MATLAB

Chunjing Mao, Yong Guan, and Yongmei Liu

Capital Normal University
mcjing@tom.com

Abstract. The methods of FM synthesis and sampling intervals synthesis can only focus on one of the two indexes which named Timbre and design-flexibility in the design of digital-synthesis. Therefore, it based on the theory of digital music, and to combining the advantages of both methods to try a new design method of digital-synthesis; and to realize the design by time-domain waveform envelope processing and frequency-domain modulation with the tool MATLAB. By the experimental analysis, Synthesis can sound not only better performance, but also have a certain flexibility in design and application; as small requirement in the size of Wavetable, also providing a reference to the designing of digital synthesizer for portable and small equipment.

1 Introduction

Nearly, all natural signals can be realized through digital technology, include music. The development of modern music-synthesis had experienced the electronic simulation, digital simulation, sampling intervals, and the physical model in four periods [1]. The methods of FM (Frequency Modulation) synthesis and sampling intervals synthesis are commonly used in digital-synthesis at present. The FM synthesis has a better flexibility in design and application but lack of sound performance; and sampling intervals synthesis has a better sound performance, however it has some limitation in design and application [2].

The digital signal processing is highly required in the design of Digital-synthesis [3]; as a tool of simulation, MATLAB had been widely used in digital signal processing applications. It takes the sound of guitar as an example try to study and propose a new method of digital-synthesis in design which combines the methods of FM synthesis and sampling intervals synthesis.

2 Theory of Digital Synthesis

Traditional analog-synthesizer produces the various frequency components of sounds through the signal generator made by electronic components, and then the user can change some of the sound characteristics in time-domain. And the digital-synthesizer realize the wave directly from digital, then convert it to an audio message. Therefore, the wave of digital synthesizer can be sampled also can be a calculated mathematically.

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 849–857, 2008.
© Springer-Verlag Berlin Heidelberg 2008

Meanwhile, in order to make the sound of synthesizer like real instruments, a usually method is to control the wave amplitude characteristics by envelope in time-domain. The most common envelope contains four basic parameters: ADSR [4] (Figure 1): Attack Rate (The forefront of voice wave rise time); Decay Rate (The attenuation after the increase); Sustain Level (For the sound continued, it is a sustained level before release, for the sound attenuation, it is the junction of decay mode and release mode); Release Rate (For the sound continued, it is the disappearance time after the definition key[4] release, for decay voices, it is the decay time after sustained level).

If it simulate the sounds of organ and string etc., continued sound mode should be chose; if simulate the sounds of piano and guitar etc., release sound mode should be chose. In actual playing, the envelope curve rate decay faster in treble than its of in mediant or bass. Therefore, the pulse time is shorter and the attenuation is faster when it plays treble. So, it should take some correction or compensation on section AD, DR, and RR.

However, it is not enough by just take the envelope to control the sound features, in order to make the tone of sound close to the real instruments, it is necessary to change the composition of the frequency components, and it can be realized by compensate, Frequency-domain transfer, filter [5] and so on.

Usually, digital-synthesizer can be divided into two categories:

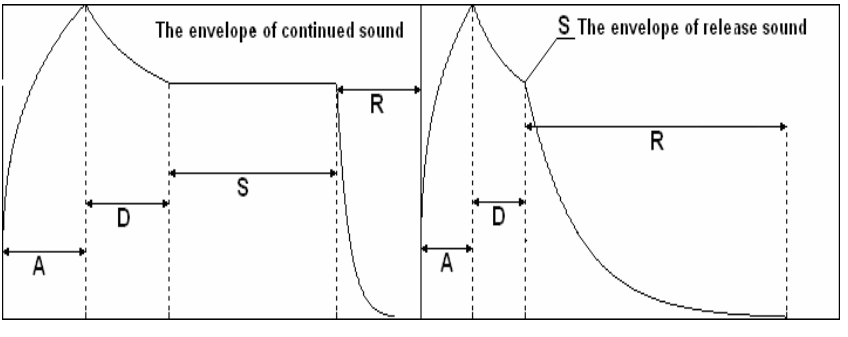


Fig. 1. The sketch of ADSR envelope in time-domain

1. FM synthesis, FM synthesis is a digital music synthesizer method developed by Stanford University in the 1970s. FM synthesis adopts frequency modulation, through some of wave (modulation signals) to change phase of the carrier frequency to realize music synthesis.

2. Sampling intervals synthesis, to compare with FM synthesis, the wave of it is not generated by the basic wave oscillator, but by the actual recording instruments playing. The theory of it is to take a certain wave as a basis, and taking amplitude processing on sampling wave to meet the requirements of playback. Then to write the wave and its parameter in ROM of the synthesizer, as the wave and its parameter are store in ROM of the synthesizer, and the synthesizer could also be called wave-table synthesizer.

3 Design and Realization

3.1 Key Period Wave Realization

It recorded the *.WAV file of single guitar play Do (sampling rate: 44.1 kHz, sampling precision: 16 bit), and convert it to a *.mat file which is the standard storage format of MATLAB, draw its time-domain wave, and zoom time-axis of section A, to see the envelope (Figure 2).

ADSR sounds of the guitar have great difference through the wave of time-domain. So, it intercept four part of key periods wave from ADSR (Figure 3).

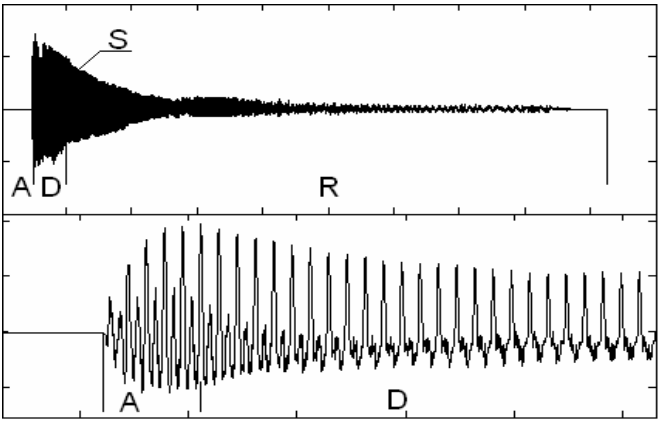


Fig. 2. The time-domain wave of guitar single sounds and the amplificatory wave of section A

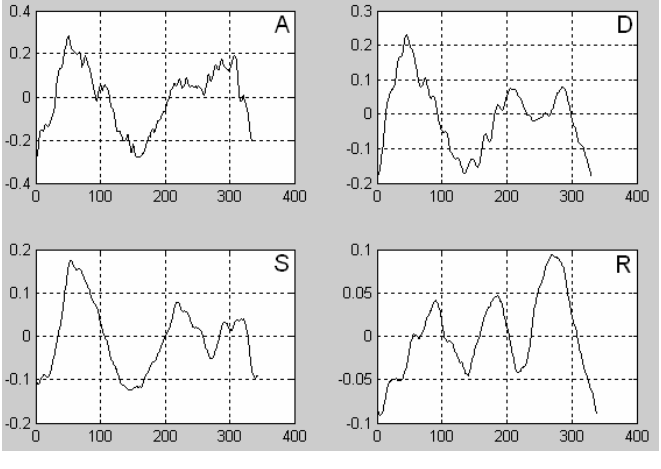


Fig. 3. The key periods of ADSR

3.2 Processing of Envelope in Time-Domain

In order to make the processing easily, it take a number of recursion on key periods of ADSR according to the actual proportion of time (see Figure 2), and then take processing of amplitude individually with the actual envelope of guitar sound in time domain.

Envelope of section A: To considered that it is a short duration while section A last, then take processing on section A with envelope of increased power function and combine with envelope of release sound mode of actual envelope from guitar sound. Codes are as follows:

```
a1=0:0.005:1;%set increased step and final value
e1=a1.^0.3;% generate Envelope with power 0.3
……
for i=1:1:200%duration of wave, calculate by sample
point
y1(i)=x1(i)*e1(i);% amplitude modulation according to
wave after recursion
end
```

Among them, x1 is the wave after recursion, e1 is the envelope function, and y1 is the final outcome of section A. Through Listening in experiments, wave duration should not be too long, otherwise there is intensive-lose in increase process, could not represent sounds feeling of burst; also it could not be too short, otherwise there will be a blasting noise as step-response.

Envelope of section D, R: actually, section D, R are processing of sound attenuation, because of human hearing reaction is based on relative changes rather than absolute changes, and the sound reaction of human ears is the process of logarithm [6]. So it modulates section D and R with envelope of index fade function, and it combine with envelope of actual sound of guitar in same sections. Codes are as follows:

```
%Section D
fp2=0.0004;%set fade parameter
a2=0:fp2:1277*fp2; %set step and final value of process
attenuation
e2=exp(-a2); %index fade, generate envelope
% Section R
fp3=0.0002; % set fade parameter
a3=2555*fp3:fp3:(l+2555)*fp3; % set step and final
value of process attenuation
e3=exp(-a3); %% Section R
……
%Section D
```

```
for j=1:1:1277% duration of wave, calculate by sample
point
y2(j)=x2(j)*e2(j); % amplitude modulation, according to
wave after recursion
end
% Section R
for k=1:1:40000% duration of wave, calculate by sample
point
y3(k)=x3(k)*e3(k); % amplitude modulation, according to
wave after recursion
end
```

Among them, x2, x3 are the waves after recursion of section D and R, e2, e3 are envelope functions, y2, y3 are ultimate outcome of section D, R. fp2, fp3 are fade parameters that decide the pace of attenuation in D and R, the greater it be, the faster the rate decay.

Envelope of section S: As the guitar sound is gradually release sound mode, S is just a junction point of D and R and its changes can be ignored, because of its amplitude changes are small and do not need to taking a envelope processing. Directly, it adjust the size and connect d and R smoothly.

Combine 4 section synthesized wave in order, and smooth the junction to synthesize basic wave of guitar, then through the process of FFT (Fast Fourier Transform) [7], can get the characteristics in frequency-domain (Figure 4).

3.3 Frequency-Domain Modulation

As the single sound wave is synthesized by key periods wave that there has a certain performance lose, in order to revert the sounds, it needs more processing on synthesis wave in frequency-domain.

3.3.1 Sub-Audio-Rate Frequency Modulation

In order to make synthesis sounds close to the actual sounds, it needs to add Sub-audio-rate frequency modulation which is less than 20 Hz frequency modulation wave on the basic synthesis wave. Directly, it could not be heard, but it can be reflected by the effects of carrier modulation.

Generally, Sub-audio-rate frequency modulation has 2 kinds: vibrato modulation and amplitude modulation. Respectively, for the production of vibrato and Tremolo; also, it need to control the depth of the vibrato [8]. The experiments show that the effect is good while vibrato in 6.4 Hz and Tremolo in 3.7 Hz.

3.3.2 Audio-Rate Frequency Modulation

Based on Sub-audio-rate frequency modulation, it needs to include the Audio-rate frequency modulation, and to control the appropriate modulation frequency, to produce sounds affluently than basic synthesis wave.

Firstly, it is necessary to introduce Audio-rate frequency modulation and its relative principles. In frequency domain, when carrier wave frequency greater than 20 Hz,

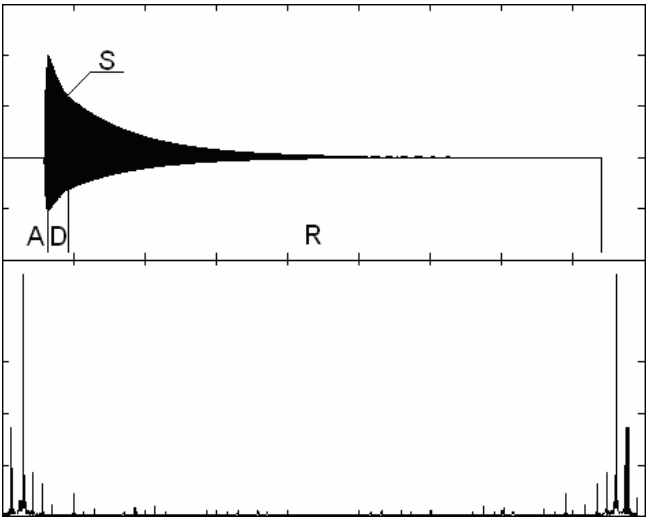


Fig. 4. The basic wave of synthesized guitar single sound and its feature in frequency-domain

sidebands [9](a kind of additional frequency) will be distribute in the carrier on both sides of the center frequency, symmetrically. A part of carrier wave energy will consumed for the sideband generation. Sideband and its energy can be accurately estimated and controlled by digital technology and the usually application method is Chowning FM algorithm, which can be simply described [10] as:

$$S_{nf} = C_f \pm nM_f \quad \text{n=0,1,2,3……} \tag{1}$$

In formula 1, S_{nf} is the sidebands frequency, C_f is the carrier wave frequency, M_f is the modulation frequency, n is the number of sidebands. When S_{nf} is a negative, in fact, the frequency has moved from 0 to 180 degrees. In the calculation, it could be dealt with absolute value, namely:

$$S_{nf} = \left| C_f \pm nM_f \right| \quad \text{n=0,1,2,3……} \tag{2}$$

While $M_f : C_f$ is a integer, or $S_{nf} : C_f$ is a integer, a special sideband - Harmonic will be produced [11]. It strengthens and enriches the sound characteristics in performance, and makes people feel keystones (first side) clearly.

As the increasing of modulation index, the sequences of sideband-modulation gradually emerging and the distribution of their amplitude obeys a Bessel function. But with the increasing of modulation index, more and more frequency ingredients could be heard, and the upside sequence may be reached with Nyquist Frequency [12]. While upside sequence reached Nyquist Frequency band, it will create some new frequency components, and for digital synthesis it is a tough problem. So, it needs to restrict the maximum modulation index to end it.

In order to make sound more affluently, a complex signal modulation with multi-signal could be considered. For audio signals, usually, 3 sinusoidal signal modulations would be sufficiently. But in actual synthesis, it is a complex algorithm with low efficiency that to Produce Harmonic by modulation. So, a direct linearity superposition of basic synthesis wave would be considered. Codes are as follows:

```
for l=1:1:44478
y(l)=x(l)+xx1(l)+xx2(l)*0.8+xx3(l)*0.4+……;% linearity
superposition
end
```

Among them, x is basic wave (carrier wave), xx1, xx2, xx3,…… are sequence for the Harmonic, y is the result of the modulation.

After the modulation in frequency domain it could be seen the characteristics of frequency are more affluent than before (Figure 5).

4 Analysis in Experiment

In recording process, it could not avoid a variety of noise to produce the glitch on the actual waveform (Figure 1). But with the comparison of the time-domain envelope to the synthesis waveform, it could be found that these two are basically accorded. There is little difference between these two waveforms in timbre, because that the key periods of the synthesis waveform are coming from the same waveform. Also, the frequency-domain characteristics (Figure 5, Figure 6) are basically consistent in both actual and synthesis waveform. Then, by the repeated listening, it could safely deem that the synthesis waveform have a good sound performance.

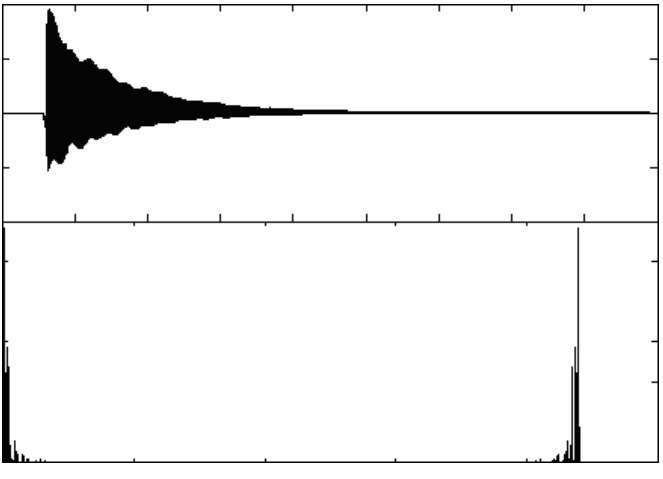


Fig. 5. The synthesized wave and its frequency-domain feature after frequency-domain modulation

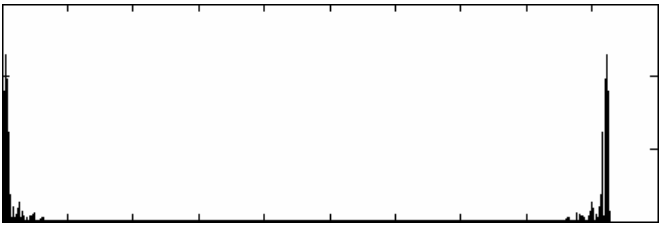


Fig. 6. The frequency-domain feature of guitar single sounds Do

Based on the single sound, to consider join the scale changes. Octave in the relations of frequency composed with twelve scale, each of the two adjacent semitones, that their frequency is $2^{1/12}$-th times from each high to low. Therefore, it can change the frequency of the synthesis sounds to makes scale on the changes. Through GUI (Graphical User Interface) in MATLAB to programming a simple playing program. Easily, it adds the control of volume, vibrato, reverb, and tone though parameters setting to implement the design flexibility.

And because of this method is based on the key periods' wave, the wavetable file are small. YAMAHA XG50, as a mainstream wavetable, which has the total size of 16497K for 128 kinds of musical instruments, the wavetable file of each instrument is about 128.88K; and the waveform generated of this method is documents only 2.81K, with control program procedures 1.17K , totally, they are 3.98K. It has a greater advantage to compare with mainstream wavetable; and it is suitable for the transplantation in embedded systems and DSPs.

5 Conclusion

In those two methods of digital-synthesis, FM synthesis has a better design flexibility, and sampling intervals synthesis has a better sound performance, but their advantages are inadequacy of each other's. It takes the sound of guitar as an example to try a new method in design of digital synthesizer by combining with the theory of digital music, and realized it in PC by MATLAB. The design has combined both advantages of FM synthesis and sampling intervals synthesis; effectively, it could reduce the storage of the wavetable.

The Design provides a reference for the DSP or embedded system in design transplantation; also, it has a certain value on the portable, small digital - synthesis system in both theory and engineering.

References

1. Wang, R.: Brief summary of the development of contemporary computer music. Musicology in China (2003)
2. Liu, Y.: The development and realization of embedded audio system. Master Degree Thesis of Anhui University (2007)
3. Sheng, Y., Zhang, Y., Zhang, X.-t.: Progress of international audio engineering. Audio Engineering 31(1), 80–83 (2007)

4. Hui, H., Li, G.: Research and development of FM synthesis. Journal of Shanxi Institute of Techology 19(2), 254–255 (2003)
5. Zeng, Y., Zhang, B., Wu, P.: Signal and System. Beijing Institute of Technology Press, Beijing (1992)
6. Li, Y., Hu, X., Zhang, Y.: Objective evaluation method of speech quality based on human auditory model. Journal of Huanzhong University of Science and Techology 28(5), 63–65 (2000)
7. Cristi, R.: Modern Digital Signal Processing. China Machine Press, Beijing (2005)
8. Hass, J.: Principles of Audio-Rate Frequency Modulation, http://www.indiana.edu/
9. Fan, C., Zhang, F., Xu, B.: Communication System Principles, 5th edn. National Defence Industry Press, Beijing (2002)
10. Chowning, J.: The synthesis of complex audio spectra by means of frequency modulation. Journal of the Audio Engineering Society 21(7) (1973)
11. Truax, B.: Organizational techniques for C:M ratios in frequency modulation. Computer Music Journal 1(4), 39–45 (1978)
12. Hass, J.: Introduction to Computer Music:Volume One, `http://www.indiana.edu/`

ERRATUM

Integrated Agent-Based Modeling with GIS for Large Scale Emergency Simulation

Dan Guo, Bo Ren, and Cheng Wang

College of Hydropower and Information Engineering,
Huazhong University of Science and Technology,
430074 Wuhan, China
guodandd@126.com

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 618–625, 2008.
© Springer-Verlag Berlin Heidelberg 2008

DOI 10.1007/978-3-540-92137-0_94

In the original version, the pages 618, 619, 622, 623 & 625 don't have the below additional information.

1. On Page 618 of this paper, the sentence "the use of ABM for experimenting and exploring geographical phenomena, specifically linking it to GIS is still in its infancy" was quoted from the paper 110 "Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations" by C. J. Castle, and A. T. Crooks, Centre for Advanced Spatial Analysis, Working Paper 110, University College London, 2006.
2. On Page 619, the sentence "ABM is a computational methodology that allows the analysts to create, analyze, and experiment with artificial worlds populated by agents that interact in nontrivial ways" was quoted from the paper "Endogenizing geopolitical boundaries with agent-based modelling" by Lars-Erik Cederman, Proceedings of the National Academy of Sciences of the United States of America, Vol. 99, No. 10, 7294-7303, 2002.
3. Page 622, Fig. 3 was taken from the paper "Constructing and Implementing an Agent-Based Model of Residential Segregation through Vector GIS" by A. T. Crooks, Centre for Advanced Spatial Analysis, Working Paper 133, University College London, 2008.
4. On Page 623, "The open circles are labeled A, V, D, and P, representing program modules which called to animate the walks (A), visualize the geometry and surfaces (V), display numeric data (D), and plot such data (P)." was quoted from the paper "Local Movement: Agent-Based Models of Pedestrian Flow" by Michael Batty, Bin Jiang, and Mark Thurstain Goodwin, Centre for Advanced Spatial Analysis, Working Paper 4, University College London, 1998.

5. On Page 625, the sentences “Currently, ABM used for simulation analysis has only been applied to abstract systems, which was designed to test how space affects the simulation outcome. Further investigation will apply the models to real world situations and to study issues in detail while testing different scenarios.” were quoted from the paper “Exploring Cities using Agent Based Models and GIS” by A. T. Crooks, Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects, 2006.

6. On Page 625, the sentence “key criteria pertaining to a selection of simulation/modeling systems, is need to be provided to facilitate the identification of a suitable system for the development of a geospatial agent-based model” was quoted from the paper “Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations” by C. J. Castle, and A. T. Crooks, Centre for Advanced Spatial Analysis, Working Paper 110, University College London, 2006.

The original online version for this chapter can be found at
http://dx.doi.org/10.1007/978-3-540-92137-0_68

Author Index

Antunes, Ana 248

Bao, Hong 562
Benli, Kristin S. 803
Bi, Wei 280

Cai, Zhenhua 125
Cai, Zhihua 510, 520, 556
Cao, Hui 501
Chen, Bo 426
Chen, Fangjie 181
Chen, Haiyang 571
Chen, Junjie 368
Chen, Qiao 479
Chen, Qing 342
Chen, Xinmeng 378
Chen, Zhifen 239
Chen, Ziyi 37
Cheng, Jingde 637
Coutinho, Maurilio Pereira 228

Dai, Guangming 162, 181
Dai, Wenjin 626
Dai, Yongtao 626
Deng, Yu 841
Diao, Yongfeng 760
Dias, Fernando Morgado 248
Ding, Jicheng 680
Ding, Lixin 669
Ding, Shifei 315
Dong, Jianli 571
Dong, Yongfeng 189
Dou, Quansheng 98
Du, Jun 591
Düzağaç, Remzi 803

El-Bakry, Hazem M. 489
Eskil, M. Taner 803

Fan, Yuanyuan 125
Fang, Pan 67, 181
Fang, Yanjun 331
Feng, Yuan 117
Feng, Yucai 546
Filgueiras, Leonardo Schilling 228
Fu, Lihua 599

Gao, Feng 322
Gao, Ge 322
Gao, Ruizhen 649
Gao, Wenxiang 322
Gao, Xiaolei 461
Gong, Chenglong 117
Gong, Wenyin 520
Goodman, Erik D. 707
Govindan, V.K. 171
Gu, Junhua 189
Guan, Jing 19
Guan, Yong 849
Guo, Dan 618
Guo, Weiya 142

Hamada, Mohamed 489
Han, Jin 470
He, Huacan 599
He, Hui 426
He, Lili 649
He, Yichao 89
Hong, Yi 659
Hu, Chengyu 750
Hu, Chenyu 67
Hu, Guangdao 740
Huang, Shitan 206
Huang, Xuejuan 378
Huang, Zhangcan 198, 461

Jeon, IlSoo 785
Jia, Weikuan 315
Jiang, Dengying 198
Jiang, Jing 470
Jiang, Liangxiao 350
Jiang, Siwei 556
Jiang, Tao 546
Jiang, Wenxia 198
Jiang, Xiao-yu 360
Jiao, Minghai 79
Jin, Jianming 426

Kang, Lishan 9, 28, 37, 206
Kang, Zhuo 28
Ke, Jinchuan 697

Khoshsima, Gholamreza 258
Kim, JongNam 778, 812
Kim, WonHee 778
Kou, Yingzhan 89
Kuang, Zhanghui 280, 715
Kuntanapreeda, Suwat 608
Kwak, Keun-Chang 220

Lai, Xinsheng 731
Lambert-Torres, Germano 228
Lee, Chia-Hoang 581
Li, Ali 530
Li, Bing 470
Li, Chao 416
Li, Chaoqun 350
Li, Chuanfeng 750
Li, Cunhe 386
Li, Haifang 368
Li, Hongfen 740
Li, Hui 125, 501, 723
Li, Huiguang 769
Li, Jiang-feng 740
Li, Jie 297
Li, Jiewei 510
Li, Jing 107
Li, Jun 760
Li, Miqing 107
Li, Nana 189
Li, Qu 212
Li, Tao 436, 446
Li, Wei 290
Li, Xiang 479
Li, Xiaoyan 297
Li, Yan 28
Li, Yanli 479
Li, Zhenhua 142, 707
Li, Zushu 831
Liang, Gang 436
Liang, Qingzhong 67, 125
Liang, Runpeng 821
Liang, Yongsheng 9
Lin, Guangming 9
Lin, Qingsheng 626
Liu, Bo 769
Liu, Chao 117
Liu, Chien-Liang 581
Liu, Guohai 306
Liu, Kangwei 386
Liu, Lijun 37
Liu, Sun Jun 436
Liu, Xiaobo 520
Liu, Xinzhe 697
Liu, Yijian 331
Liu, Yong 1, 58
Liu, Yongmei 849
Liu, Yujuan 189
Lu, Chengwu 687
Lu, Wei 537
Lu, Xin 9
Luo, Biao 107
Luo, Jiaxiang 821
Luo, Wenfeng 821
Lv, Hui 152

Ma, Weimu 269
Mao, Chunjing 849
Mao, Jiali 760
Martins, Helga Gonzaga 228
Marusak, Piotr M. 396
Mathew, Abraham T. 171
Mei, Congli 306
Men, Lihuan 368
Meng, Xianyu 562
Moon, KwangSeok 778, 812

Nozaki, Tsutomu 470

Peng, Lei 162
Peter, J. Dinesh 171

Qi, Xing 212
Qiang, Wei 501
Qin, Renchao 446

Ren, Bo 618
Ryu, TaeKyung 812

Salomon, Camila Paes 228
Shan, Dandan 841
Shao, Guifang 831
Shao, Yuxiang 342
Shen, Chunpu 89
Shi, Chen 206
Shi, Jie 546
Shi, Zhongzhi 98
Song, Hongtao 680
Song, Jun 134, 795
Su, Chunyang 315
Sun, Donghuai 571
Sun, Ming 680
Sun, Wu 426

Tan, Yi 181
Tang, Jiafu 79
Tian, Shengwen 530
Tong, Hengjian 537
Traore, Issa 537

Wang, Ao 239
Wang, Cheng 618
Wang, Dian Gang 436
Wang, Ercheng 649
Wang, Guan 697
Wang, Haijun 841
Wang, Jiahai 280, 715
Wang, Jianzhou 571
Wang, Li 841
Wang, Lianguo 659
Wang, Maocai 795
Wang, Weihong 212
Wang, Yilei 530
Wang, Yuanzhen 162, 546
Wang, Yuqiao 290
Wei, Yang 378
Wen, Yuhua 831
Wong, Tim 142
Wu, Jia 350
Wu, Jie 45
Wu, Jun 152
Wu, Tihua 769

Xie, Yuanhua 470
Xiong, Yuhong 426
Xu, Guangmei 562
Xu, Haixia 306
Xu, Jiazhen 378
Xu, Xinzheng 315
Xue, Chongsheng 591
Xue, Siqing 45
Xue, Zhi Bin 454

Yan, Jia 591
Yan, Xuesong 67, 206, 723, 750
Yang, Fan 134, 795
Yang, Hongyong 530
Yang, Ming 19
Yang, Qinghua 591
Yang, Qingqiang 821
Yang, Xuefeng 290
Yang, Yeon-Mo 785
Yao, Xin 9
Ye, Guo 290
Yi, Chenfu 269
Yin, Jianping 406
Yin, Xing 760
Yu, Dongmei 659
Yu, Erkeng 98
Yu, Fang 831
Yu, Jinghu 669
Yu, Zhijun 98
Yuan, Jingling 297

Zeng, Jian Chao 454
Zhan, Enqi 461
Zhan, Yubin 406
Zhang, Bin 546
Zhang, Dongmei 239, 342
Zhang, Guomin 406
Zhang, Huanguo 134, 795
Zhang, Jingjun 649
Zhang, Liwen 315
Zhang, Ming 117
Zhang, Qingbin 769
Zhang, Ying 760
Zhang, Yu 446
Zhang, Yunong 269
Zhao, Dan 142
Zhao, Fuqing 659
Zhao, Kui 436
Zhao, Lin 680
Zhao, Ling 426
Zheng, Jianbin 461
Zheng, Jinhua 107, 152
Zheng, Yongzhi 98
Zhong, Luo 297
Zhou, Dawei 306
Zhou, Keping 322
Zhou, Ying 715
Zhou, Zuan 181
Zhu, En 406
Zhu, Jian 350
Zhu, Lina 386
Zhu, Tong 470