Springer Proceedings in Mathematics & Statistics

Alexey Sorokin
Robert Murphey
My T. Thai
Panos M. Pardalos *Editors*

Dynamics of Information Systems: Mathematical Foundations

Springer

Springer Proceedings in Mathematics & Statistics

Volume 20

For further volumes:
http://www.springer.com/series/10533

Springer Proceedings in Mathematics & Statistics

This book series will feature volumes of selected contributions from workshops and conferences in all areas of current research activity in mathematics and statistics, operations research and optimization. In addition to an overall evaluation of the interest, scientific quality, and timeliness of each proposal at the hands of the publisher, every individual contribution is refereed to standards comparable to those of leading journals in the field. This expanded series thus proposes to the research community well-edited and authoritative reports on newest developments in the most interesting and promising areas of mathematical and statistical research today.

Alexey Sorokin • Robert Murphey • My T. Thai
Panos M. Pardalos
Editors

Dynamics of Information Systems: Mathematical Foundations

Springer

Editors
Alexey Sorokin
Industrial and Systems Engineering
University of Florida
Gainesville, FL
USA

My T. Thai
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL
USA

Robert Murphey
Air Force Research Lab
Munitions Directorate
Eglin Air Force Base, FL
USA

Panos M. Pardalos
Center for Applied Optimization
Industrial and Systems Engineering
University of Florida
Gainesville, FL
USA

Laboratory of Algorithms and Technologies for Networks Analysis (LATNA)
National Research University
Higher School of Economics
Moscow, Russia

ISBN 978-1-4614-3905-9 ISBN 978-1-4614-3906-6 (eBook)
DOI 10.1007/978-1-4614-3906-6
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2012939429

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Information systems have become an inevitable part of contemporary society and affect our lives every day. With rapid development of the technology, it is crucial to understand how information, usually in the form of sensing and control, influences the evolution of a distributed or networked system, such as social, biological, genetic, and military systems. The dynamic aspect of information fundamentally describes the potential influence of information on the system and how that information flows through the system and is modified in time and space. Understanding this dynamics will help to design a high-performance distributed system for real-world applications. One notable example is the integration of sensor networks and transportation where the traffic and vehicles are continuously moving in time and space. Another example would be applications in the cooperative control systems, which have a high impact on our society, including robots operating within a manufacturing cell, unmanned aircraft in search and rescue operations or military surveillance and attack missions, arrays of microsatellites that form distributed large aperture radar, or employees operating within an organization. Therefore, concepts that increase our knowledge of the relational aspects of information as opposed to the entropic content of information will be the focus of the study of information systems dynamics in the future.

This book presents the state of the art relevant to the theory and practice of the dynamics of information systems and thus lays a mathematical foundation in the field. The first part of the book provides a discussion about evolution of information in time, adaptation in a Hamming space, and its representation. This part also presents an important problem of optimization of information workflow with algorithmic approach, as well as integration principle as the master equation of the dynamics of information systems. A new approach for assigning task difficulty for operators during multitasking is also presented in this part. Second part of the book analyzes critical problems of information in distributed and networked systems. Among the problems discussed in this part are sensor scheduling for space object tracking, randomized multidimensional assignment, as well as various network problems and solution approaches. The dynamics of climate networks and complex network models are also discussed in this part. The third part of the book

provides game-theoretical foundations for dynamics of information systems and considers the role of information in differential games, cooperative control, protocol design, and leader with multiple followers games.

We gratefully acknowledge the financial support of the Air Force Research Laboratory and the Center for Applied Optimization at the University of Florida. We thank all the contributing authors and the anonymous referees for their valuable and constructive comments that helped to improve the quality of this book. Furthermore, we thank Springer Publisher for making the publication of this book possible.

Gainesville, FL, USA

Alexey Sorokin
Robert Murphey
My T. Thai
Panos M. Pardalos

Contents

Contributors

Roman V. Belavkin Middlesex University, London, UK

Pia Berg-Yuen Air Force Research Lab, Munitions Directorate, Eglin AFB, FL, USA

Vladimir Boginski Department of Industrial and Systems Engineering, University of Florida, Shalimar, FL, USA

Amélia C.D. Caldeira Departamento de Matemática, Instituto Superior de Engenharia do Porto, Porto, Portugal

Laura C. Carpi Civil, Surveying and Environmental Engineering, The University of Newcastle, New South Wales, Australia, Departamento de Física, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Genshe Chen I-Fusion Technology, Inc., Germantown, MD, USA

Huimin Chen University of New Orleans, Department of Electrical Engineering, New Orleans, LA, USA

Jess W. Curtis Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL, USA

Egidio D'Amato Dipartimento di Scienze Applicate, Università degli Studi di Napoli "Parthenope", Centro Direzionale di Napoli, Napoli, Italy

Elia Daniele Dipartimento di Ingegneria Aerospaziale, Università degli Studi di Napoli "Federico II", Napoli, Italy

Swaroop Darbha Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA

Warren E. Dixon Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA

Paola Festa Department of Mathematics and Applications, University of Napoli FEDERICO II, Compl. MSA, Napoli, Italy

Dalila B.M.M. Fontes LIAAD - INESC Porto L.A. and Faculdade de Economia, Universidade do Porto, Porto, Portugal

Fernando A.C.C. Fontes ISR Porto and Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

Michael J. Hirsch Raytheon Company, Intelligence and Information Systems, Annapolis Junction, MD, USA

Alla Kammerdiner New Mexico State University, Las Cruces, NM, USA

Galina Korotkikh School of Information and Communication Technology, CQUniversity, Mackay, Queensland, Australia

Victor Korotkikh School of Information and Communication Technology, CQUniversity, Mackay, Queensland, Australia

Pavlo Krokhmal Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA, USA

Lina Mallozzi Dipartimento di Matematica e Applicazioni, Università degli Studi di Napoli "Federico II", Napoli, Italy

Mohammad Mirghorbani Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA, USA

Robert Murphey Air Force Research Lab, Munitions Directorate, Eglin AFB, FL, USA

Harsha Nagarajan Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA

Rakesh Nagi University at Buffalo, Department of Industrial and Systems Engineering, Buffalo, NY, USA

Héctor Ortiz-Peña CUBRC, Buffalo, NY, USA

Meir Pachter Air Force Institute of Technology, AFIT, Wright Patterson AFB, OH, USA

Panos M. Pardalos Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

Eduardo L. Pasiliao Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL, USA

Giovanni Petrone Dipartimento di Ingegneria Aerospaziale, Università degli Studi di Napoli "Federico II", Napoli, Italy

Khanh Pham Air Force Research Laboratory, Space Vehicles Directorate, Kirtland Air Force Base, NM, USA

Jose C. Principe Computational NeuroEngineering Laboratory, University of Florida, Gainesville, FL, USA

Kumbakonam Rajagopal Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA

Sivakumar Rathinam Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA

Martín Gómez Ravetti Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Osvaldo A. Rosso Chaos & Biology Group, Instituto de Cálculo, Universidad de Buenos Aires, Argentina, Departamento de Física, Universidade Federal de Minas Gerais, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Patricia M. Saco Civil, Surveying and Environmental Engineering, The University of Newcastle, New South Wales, Australia

John M. Shea Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA

Dan Shen I-Fusion Technology, Inc., Germantown, MD, USA

Adam Stotz CUBRC, Buffalo, NY, USA

Sankrith Subramanian Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA

Moises Sudit CUBRC, Buffalo, NY, USA

Mujahid N. Syed Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

Simone Tancredi Dipartimento di Ingegneria Aerospaziale, Università degli Studi di Napoli "Federico II", Napoli, Italy

Alexander Veremyev Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

Part I
Evolution and Dynamics of Information Systems

Dynamics of Information and Optimal Control of Mutation in Evolutionary Systems

Roman V. Belavkin

Abstract Evolutionary systems are used for search and optimization in complex problems and for modelling population dynamics in nature. Individuals in populations reproduce by simple mechanisms, such as mutation or recombination of their genetic sequences, and selection ensures they evolve in the direction of increasing fitness. Although successful in many applications, evolution towards an optimum or high fitness can be extremely slow, and the problem of controlling parameters of reproduction to speed up this process has been investigated by many researchers. Here, we approach the problem from two points of view: (1) as optimization of evolution in time; (2) as optimization of evolution in information. The former problem is often intractable, because analytical solutions are not available. The latter problem, on the other hand, can be solved using convex analysis, and the resulting control, optimal in the sense of information dynamics, can achieve good results also in the sense of time evolution. The principle is demonstrated on the problem of optimal mutation rate control in Hamming spaces of sequences. To facilitate the analysis, we introduce the notion of a relatively monotonic fitness landscape and obtain general formula for transition probability by simple mutation in a Hamming space. Several rules for optimal control of mutation are presented, and the resulting dynamics are compared and discussed.

Keywords Fitness • Information • Hamming space • Mutation rate • Optimal evolution

R.V. Belavkin (✉)
Middlesex University, London NW4 4BT, UK
e-mail: R.Belavkin@mdx.ac.uk

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_1,

1 Introduction

Dynamical systems have traditionally been considered as time evolution using mathematical models based on Markov processes and corresponding differential equations. These methods achieved tremendous success in many applications, particularly in optimal estimation and control of linear and some non-linear systems [6,12,18]. Markov processes have also been applied in studies of learning [11,20,21] and evolutionary systems [2, 14, 22]. Their optimization, however, is complicated for several reasons. One of them is that the relation between available controls and values of an objective function is not well defined or uncertain. Another is an incredible complexity associated with their optimization.

The first difficulty can be sometimes overcome either by defining and analysing the underlying structure of the system or by learning the relationships between the controls and objective function from data. Here, we take the former approach. We first outline some general principles by relating a topology of the system to the objective function. Then we consider probability of simple mutation of sequences in a Hamming space, and derive expressions for its relation to values of a fitness function. The resulting system, however, although completely defined, quickly becomes intractable for optimization of its evolution in time using traditional methods with the exception of a few special cases.

Evolution of dynamical systems can be considered from another point of view as evolution in information. In fact, dynamic information is one of the main characteristics of learning and evolutionary systems. Information dynamics can be understood simply as changes of information distance between states, represented by probability measures on a phase space. Although optimality with respect to information has been studied in theories of information utility [19] and information geometry [1, 8], there were few attempts to integrate information dynamics in synthesis of optimal control of dynamical systems [3–5, 7]. Understanding better the relation between optimality with respect to time and information criteria has been the main motivation for this work.

In the next section, we formulate and consider problems of optimization of evolution in time and in information. Then we consider evolution of a discrete system of sequences and derive relevant expressions for optimization of their position in a Hamming space. Special cases will be considered in Sect. 4 to derive several control functions for mutation rate and evaluate their performance. Then we shall summarize and discuss the results.

2 Evolution in Time and Information

Let Ω be the set of elementary events and $f : \Omega \to \mathbb{R}$ be an objective function. An evolution of a system is represented by a sequence $\omega_0, \omega_1, \ldots, \omega_t, \ldots$ of events, indexed by time, and we shall consider a control problem optimizing the evolution

with respect to the objective function. For simplicity, we shall assume that Ω is countable or even a finite set, so that there is at most a countable or finite number of values $x = f(\omega)$. This is because we focus in this paper on applications of such problems to biological or evolutionary systems. In this context, Ω represents the set of all possible individual organisms (e.g. the set of all DNA sequences), and f is called a *fitness* function. The sequence $\{\omega_t\}$ represents descendants of ω_0 in $t \geq 0$ generations.

Fitness function represents (or induces) a total pre-order $\lesssim$ on Ω: $a \lesssim b$ if and only if $f(a) \leq f(b)$. It factorizes Ω into the equivalence classes of fitness:

$$[x] := \{\omega \in \Omega : f(\omega) = x\}.$$

Thus, from the point of optimization of fitness f, sequences $\omega_0, \ldots, \omega_t, \ldots$ corresponding to the same sequence $x_0, \ldots, x_t, \ldots$ of fitness values are equivalent. Equivalent evolutions are represented by the real stochastic process $\{x_t\}$ of fitness values.

2.1 Optimization of Evolution in Time

Let $P(x_{s+1} \,|\, x_s)$ be the conditional probability of an offspring having fitness value $x_{s+1} = f(\omega_{s+1})$ given that its parent had fitness value $x_s = f(\omega_s)$. This Markov probability can be represented by a left stochastic matrix T, and if transition probabilities $P(x_{s+1} \,|\, x_s)$ do not depend on s, then T defines a stationary (or time-homogeneous) Markov process x_t. In particular, T^t defines a linear transformation of distribution $p_s := P(x_s)$ of fitness values at time s into distribution $p_{s+t} := P(x_{s+t})$ of fitness values after $t \geq 0$ generations:

$$p_{s+1} = T p_s = \sum_{x_s \in f(\Omega)} P(x_{s+1} \mid x_s)\, P(x_s)\,, \quad \Rightarrow \quad p_{s+t} = T^t p_s.$$

The expected fitness of the offspring after t generations is

$$\mathbb{E}\{x_{s+t}\} := \sum_{x_{s+t} \in f(\Omega)} x_{s+t}\, P(x_{s+t}).$$

We say that individuals adapt if and only if $\mathbb{E}\{x_{s+t}\} \geq \mathbb{E}\{x_s\}$.

Suppose that the transition probability $P_\mu(x_{s+1} \mid x_s)$ depends on a control parameter μ, so that the Markov operator $T_{\mu(x)}$ depends on the control function $\mu(x)$. Then the expected fitness $\mathbb{E}_{\mu(x)}\{x_{s+t}\}$ also depends on $\mu(x)$. In the context of biological or evolutionary systems, μ can be related to a reproduction strategy, which involves mutation and recombination of DNA sequences. The optimal control should maximize expected fitness of the offspring to achieve maximum or fastest adaptation. This problem, however, can be formulated and solved in different ways.

Optimality at a certain generation is defined by the following (instantaneous) optimal value function:

$$\overline{f}(\lambda) := \sup_{\mu(x)} \{\mathbb{E}_{\mu(x)}\{x_{s+t}\} : t \leq \lambda\}. \tag{1}$$

Here, $\lambda \geq 0$ represents a time constraint. Function $\overline{f}(\lambda)$ is non-decreasing, and optimization problem (1) has dual representation by the inverse function

$$\overline{f}^{-1}(\upsilon) := \inf_{\mu(x)} \{t \geq 0 : \mathbb{E}_{\mu(x)}\{x_{s+t}\} \geq \upsilon\}. \tag{2}$$

Here, υ is a constraint on the expected fitness at $s + t$. Thus, $\overline{f}(\lambda)$ is defined as the maximum adaptation in no more than λ generations; $\overline{f}^{-1}(\upsilon)$ is defined as the minimum number of generations required to achieve adaptation υ. Observe that $\overline{f}(\lambda)$ can in general have infinite values, and we can define $\overline{f}(\infty) := \sup f(\omega)$. However, $\overline{f}^{-1}(\sup f(\omega)) \leq \infty$.

Observe that optimal solutions $\mu(x)$ to problems (1) or (2) depend on the constraints λ or υ (and on the initial distribution p_s via $T^t p_s = p_{s+t}$). If the objective is to derive one optimal function $\mu(x)$ that can be used throughout the entire "evolution" $[s, s + t]$, then one can define another (cumulative) optimal value function

$$\overline{F}(s,t) := \sup_{\mu(x)} \sum_{\lambda=0}^{t} \mathbb{E}_{\mu(x)}\{x_{s+\lambda}\}. \tag{3}$$

This optimization problem can be formulated as a recursive series of one-step maximizations using the dynamic programming approach [6]. Also, using definitions (1) and (3), one can easily show the following inequality:

$$\overline{F}(s,t) \leq \sum_{\lambda=s}^{t} \overline{f}(\lambda).$$

Given a control function $\mu(x)$ and the corresponding operator $T_{\mu(x)}$, one can compute $\mathbb{E}_{\mu(x)}\{x_{s+t}\}$ for any fitness function $f(\omega)$ and initial distribution $p_s := P(x_s)$ of its values. Observe also that this formulation uses only the values of fitness, and therefore function $f(\omega)$ may change on $[s, s + t]$. Solving optimization problems (1) and (3), however, is not as straightforward, because it requires the inversion of the described computations.

Because we are interested in optimal μ as a function of x_s, we can take $p_s = \delta_{x_s}(x)$, and the optimal function $\mu(x)$ is given by maximizing conditional expectation $\mathbb{E}_\mu\{x_{s+t} \mid x_s\}$ for each x_s. When $P_\mu(x_{s+t} \mid x_s)$ depends sufficiently smoothly on μ, the necessary condition of optimality in problems (1) or (2) can be expressed using conditional expectations for each x_s:

$$\frac{\mathrm{d}}{\mathrm{d}\mu}\mathbb{E}_\mu\{x_{s+t} \mid x_s\} = \sum_{x_{s+t} \in f(\Omega)} x_{s+t} \frac{\mathrm{d}}{\mathrm{d}\mu} P_\mu(x_{s+t} \mid x_s) = 0. \tag{4}$$

If $\mathbb{E}_{\mu(x)}\{x_{t+s}\}$ is a concave functional of $\mu(x)$, then the above condition is also sufficient. In addition, if the optimal value function $\overline{f}(\lambda)$ is strictly increasing, then $t = \lambda$. Unfortunately, in the general case, analytical expressions are either not available or are extremely complex, and only approximate solutions can be obtained using numerical or evolutionary techniques. One useful technique is based on absorbing Markov chains and minimization of their convergence time.

Recall that a Markov chain is called *absorbing* if $P(x_{s+1} = \upsilon \mid x_s = \upsilon) = 1$ for some states υ. Such states are also called absorbing, while other states are called transient. If there are n absorbing and l transient states, then the corresponding right stochastic matrix T' (transposed of T) can be written in the canonical form to compute its *fundamental matrix* N:

$$T' = \begin{pmatrix} I_n & 0 \\ R & Q \end{pmatrix}, \qquad N = (I_l - Q)^{-1}.$$

Here, I_n is the $n \times n$ identity matrix representing transition probabilities between absorbing states; Q is the $l \times l$ matrix of transition probabilities between transient states; R is the $l \times n$ matrix of transition probabilities from transient to absorbing states; 0 is the $n \times l$ matrix of zeros (probabilities of escaping from absorbing states). The sum of elements n_{ij} of the fundamental matrix N in ith row gives the expected time $t_i = \sum_j n_{ij}$ before the process converges into an absorbing state starting in state i. Thus, given distribution $p_s := P(i)$ of states at time moment s, the expected time to converge into any absorbing state can be computed as follows:

$$\mathbb{E}\{t\} = \sum_{i=1}^{l} t_i \, P(i) = \sum_{i=1}^{l} \sum_{j=1}^{l} n_{ij} \, P(i). \tag{5}$$

The quantity above can facilitate numerical solutions to problem (1). Indeed, this problem is represented dually by problem (2) with constraint $\mathbb{E}\{x_{s+t}\} \geq \upsilon$, and one can assume states $x \geq \upsilon$ as absorbing. Then, given control function $\mu(x)$ and corresponding operator $T_{\mu(x)}$, one can compute the expected time $\mathbb{E}_{\mu(x)}\{t\}$ of convergence into the absorbing states. For example, we shall consider $\mathbb{E}_{\mu(x)}\{t\}$ for a single absorbing state $\upsilon = \sup f(\omega)$. Minimization of $\mathbb{E}_{\mu(x)}\{t\}$ over some family of control functions $\mu(x)$ can be performed numerically.

2.2 Optimal Evolution in Information

We have considered evolution on Ω as transformations $T^t : p_s \mapsto T^t p_s$ of probability measures $p_s := P(x)$ on values $x = f(\omega)$. These transformations are endomorphisms $T^t : \mathcal{P}(X) \to \mathcal{P}(X)$ of the simplex

$$\mathcal{P}(X) := \{p \in \mathcal{M}(X) : p \geq 0\,, \ \|p\|_1 = 1\}$$

of all probability measures $p := P(x)$ on $X = f(\Omega)$. Here, $\mathcal{M}(X)$ is the Banach space of all real Radon measures with the norm of absolute convergence $\|p\|_1 := \sum |P(x)|$.

Observe that expected value $\mathbb{E}\{x\} = \sum x\, P(x)$ of $x = f(\omega)$ is a linear functional $f(p) = \langle f, p\rangle$ on $\mathcal{P}(X)$. Here, f is an element of the dual space $\mathcal{M}'(X)$ with respect to the pairing $\langle\cdot,\cdot\rangle$, defined by the sum $\sum xy$. Therefore, $\mathbb{E}\{x_{s+t}\} = \langle f, p_{s+t}\rangle \geq \upsilon$ is a linear constraint in problem (2). It is attractive to consider problems (1)–(3) as linear or convex optimization problems. In theory, this can be done if one defines time-valued distance between arbitrary points in $\mathcal{P}(X)$ as follows:

$$t(p,q) := \inf_{\mu} \left\{ t \geq 0 : p = T_\mu^t\, q \right\},$$

where minimization is over some family T_μ of linear endomorphisms of $\mathcal{P}(X)$ (i.e. some family of left stochastic matrices T_μ). Then problem (1) can be expressed as maximization of linear functional $\langle f, p\rangle$ subject to constraint $t(p,q) \leq \lambda$. The computation of $t(p,q)$, however, is even more demanding than optimization problem (2) we would like to solve.

On the other hand, there exist a number of information distances $I(p,q)$ on $\mathcal{P}(X)$, which are easily computable. For example, the total variation and Fisher's information metrics are defined as follows [8]:

$$I_V(p,q) := \sum_{x\in f(\Omega)} |P(x) - Q(x)|\,, \quad I_F(p,q) := 2\arccos \sum_{x\in f(\Omega)} \sqrt{P(x)Q(x)}.$$

Another important example is the Kullback–Leibler divergence [13]:

$$I_{\mathrm{KL}}(p,q) := \sum_{x\in f(\Omega)} \left[\ln \frac{P(x)}{Q(x)}\right] P(x). \tag{6}$$

It has a number of important properties, such as additivity $I_{\mathrm{KL}}(p_1 p_2, q_1 q_2) = I_{\mathrm{KL}}(p_1, q_1) + I_{\mathrm{KL}}(p_2, q_2)$, and optimal evolution in I_{KL} is represented by an evolution operator [5]. Thus, given an information distance $I : \mathcal{P}\times\mathcal{P} \to \mathbb{R}_+ \cup \{\infty\}$, we can define the following optimization problem:

$$\overline{\phi}(\lambda) := \sup_p \{\mathbb{E}_p\{x\} : I(p,q) \leq \lambda\}. \tag{7}$$

Here, λ represents an information constraint. Problem (7) has dual representation by the inverse function

$$\overline{\phi}^{\,-1}(\upsilon) := \inf_p \{I(p,q) : \mathbb{E}_p\{x\} \geq \upsilon\}. \tag{8}$$

These problems, unlike (1) and (2), have exact analytical solutions, if $I(p,q)$ is a closed (lower semicontinuous) function of p with finite values on some neighbourhood in $\mathcal{P}(X)$. For example, the necessary and sufficient optimality

conditions in problem (7) are expressed using the Legendre–Fenchel transform $I^*(f,q) := \sup_p[\langle f,p\rangle - I(p,q)]$ of $I(\cdot,q)$, and can be obtained using the standard method of Lagrange multipliers (see [4] for derivation). In particular, if $I^*(\cdot,q)$ is Gâteaux differentiable, then $p(\beta)$ is an optimal solution if and only if:

$$p(\beta) = \nabla I^*(\beta f,q)\,, \quad I(p(\beta),q) = \lambda\,, \quad \beta^{-1} = \mathrm{d}\overline{\phi}(\lambda)/\mathrm{d}\lambda\,, \quad \beta^{-1} > 0. \tag{9}$$

Here, $\nabla I^*(\cdot,q)$ denotes gradient of convex function $I^*(\cdot,q)$. For example, the dual functional of $I_{\mathrm{KL}}(p,q)$ is

$$I^*_{\mathrm{KL}}(f,q) := \ln \sum_{x\in f(\Omega)} \mathrm{e}^x\, Q(x). \tag{10}$$

Substituting its gradient into conditions (9), one obtains optimal solutions to problems (7) or (8) as a one-parameter exponential family:

$$p(\beta) = \mathrm{e}^{\beta f - \Psi_f(\beta)}\, p(0)\,, \quad p(0) = q, \tag{11}$$

where $\Psi_f(\beta) := \ln I^*_{\mathrm{KL}}(\beta f,q)$ is the cumulant generating function. Its first derivative $\Psi'_x(\beta)$, in particular, is the expected value $\mathbb{E}_{p(\beta)}\{x\} = \langle f, p(\beta)\rangle$. Equation (11) corresponds to the following differential equation:

$$p'(\beta) = [f - \langle f, p(\beta)\rangle]\, p(\beta). \tag{12}$$

This is the *replicator equation*, studied in population dynamics [15]. Note that fitness function $f(\omega)$, defining the replication rate, may depend on p in a general case. One can see that optimal evolution in information divergence I_{KL} corresponds to replicator dynamics with respect to parameter β—the inverse of a Lagrange multiplier related to the information constraint λ as $\beta^{-1} = \mathrm{d}\overline{\phi}(\lambda)/\mathrm{d}\lambda$. This property is unique to information divergence I_{KL} [5], and we shall focus in this paper on optimal evolution (11).

3 Evolution of Sequences

The main object of study in this work is a discrete system Ω of sequences representing biological or artificial organisms. Such systems, although finite, can be too large to enumerate on a digital computer, and there are an infinite number of possible evolutions of finite populations of the organisms. It is possible to factorize the system by considering the evolution only on the equivalence classes, defined by an objective function, which is what we have described in previous section. The difficulty, however, is understanding the relation between the controls, which act on and transform elements of Ω, and the factorized system $\Omega/\!\sim$. In this section, we make general considerations of this issue, and then derive specific equations for the case, when Ω is a Hamming space of sequences.

3.1 Topological Considerations and Controls

We have defined the problem of optimal control of evolution of events in Ω as a Markov decision process, where $P_\mu(x_{s+1} \mid x_s)$ is the transition probability between different values $x = f(\omega)$ of the objective function and depending on the control parameter μ. The specific expression for $P_\mu(x_{s+1} \mid x_s)$ depends on the structure of the domain Ω of the objective function and the range of possible controls μ. If the control μ is a kind of a search operator in Ω, then a structure on Ω can facilitate the search process.

Recall that Ω is a totally pre-ordered set: $a \lesssim b$ if and only if $f(a) \leq f(b)$. The structure on Ω must be reach enough to embed this pre-order. For example, if τ is a topology on Ω, then it is desirable that the principal downsets $\downarrow a := \{\omega \lesssim a\}$ are closed in τ, while their complements $\Omega \setminus \downarrow a$ are open. Indeed, a sequence $\omega_0, \ldots, \omega_s, \ldots$ such that $\omega_{s+1} \in \Omega \setminus \downarrow \omega_s$ corresponds to a sequence of strictly increasing values $x_i = f(\omega_i)$. If there exists an optimal (top) element $\top \in \Omega$ such that $\sup f(\omega) = f(\top)$, then such a sequence converges to $x = f(\top)$. Note that finite set Ω always contains $\top$ and $\bot$ elements.

Let us define the following property of the objective function $f(\omega)$, which will also clarify the terms "smooth" and "rugged" fitness landscape, used in biological literature. Let us equip Ω with a metric $d : \Omega \times \Omega \to [0, \infty)$, so that similarity between a and $b \in \Omega$ can be measured by $d(a, b)$. We define f to be locally monotonic relative to d.

Definition 1 (Monotonic landscape). Let (Ω, d) be a metric space, and let $f : \Omega \to \mathbb{R}$ be a function with $f(\top) = \sup f(\omega)$ for some $\top \in \Omega$. We say that f is *locally monotonic* (*locally isomorphic*) relative to metric d if for each $\top$ there exists a ball $B(\top, r) := \{\omega : d(\top, \omega) \leq r\} \neq \{\top\}$ such that for all $a, b \in B(\top, r)$:

$$-d(\top, a) \leq -d(\top, b) \quad \Longrightarrow (\Longleftrightarrow) \quad f(a) \leq f(b).$$

We say that f is *monotonic* (*isomorphic*) relative to d if $B(\top, r) \equiv \Omega$.

Example 1 (Negative distance). If f is isomorphic to d, then one can replace $f(\omega)$ by the negative distance $-d(\top, \omega)$. The number of values of such f is equal to the number of spheres $S(\top, r) := \{\omega : d(\top, \omega) = r\}$. One can easily show also that when f is isomorphic to d, then there is only one $\top$ element: $f(\top_1) = f(\top_2) \iff d(\top_2, \top_1) = d(\top_2, \top_2) = 0 \iff \top_1 = \top_2$.

Example 2 (Needle in a haystack). Let $f(\omega)$ be defined as

$$f(\omega) = \begin{cases} 1 & \text{if } d(\top, \omega) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

This function is often used in studies of performance of genetic algorithms (GAs). In biological literature, $\top$ element is often referred to as the *wild type*, and a two-valued landscape is used to derive error threshold and critical mutation rate [15].

One can check that if for each $\top \in \Omega$ there exists $B(\top, r) \neq \{\top\}$ containing only one $\top$, then two-valued f is locally monotonic relative to any metric. Indeed, conditions of the definition above are satisfied in all such $B(\top, r) \subset \Omega$. If Ω has unique $\top$, then the conditions are satisfied for $B(\top, \infty) = \Omega$. Optimal function $\mu(x)$ for such $f(\omega)$ is related to maximization of probability $P_\mu(x_{s+1} = 1 \mid x_s)$.

For monotonic f, spheres $S(\top, l)$ cannot contain elements with different values $x = f(\omega)$. We can generalize this property to *weak* or ϵ-monotonicity, which requires that the variance of $x = f(\omega)$ within elements of each sphere $S(\top, l)$ is small or does not exceed some $\epsilon \geq 0$. These assumptions allow us to replace $f(\omega)$ by negative distance $d(\top, \omega)$ and derive expressions for transition probability $P_\mu(x_{s+1} \mid x_s)$ using topological properties of (Ω, d).

Monotonicity of f depends on the choice of metric, and one can define different metrics on Ω. Generally, one prefers metric d_2 to d_1 if the neighbourhoods, where f is monotonic relative to d_2, are "larger" than for metric d_1: $B_1(\top, r) \subseteq B_2(\top, r)$ for all $B_i(\top, r)$, where f is monotonic relative to d_i. In this respect, the least preferable is the discrete metric: $d(a, b) = 0$ if $a = b$; 0 otherwise. We shall now consider the example of Ω being the Hamming space, which plays an important role in theoretical biology as well as engineering problems.

3.2 Mutation and Adaptation in a Hamming Space

Biological organisms are represented by DNA sequences, and reproduction involves mutation and recombination of the parent sequences. Generally, a set of sequences Ω can be equipped with different metrics and topologies. Here, we shall consider the case when Ω is a Hamming space $\mathcal{H}_\alpha^l := \{1, \ldots, \alpha\}^l$—a space of sequences of length l and α letters and equipped with the Hamming metric $d(a, b) := |\{i : a_i \neq b_i\}|$. We shall also consider only asexual reproduction by simple mutation, which is defined as a process of independently changing each letter in a parent sequence to any of the other $\alpha - 1$ letters with probability $\mu/(\alpha - 1)$. This point mutation is defined by one parameter μ, called the *mutation rate*. Assuming that fitness function $f(\omega)$ is isomorphic to the negative Hamming distance $d(\top, \omega)$, we shall derive probability $P_\mu(x_{s+1} \mid x_s)$ and optimize evolution of sequences by controlling the mutation rate. This complex problem has relevance not only for engineering problems but also for biology, because the abundance of neutral mutations in nature supports an intuition that biological fitness landscapes are at least weakly locally monotonic relative to the Hamming metric.

We analyse asexual reproduction by mutation in metric space $\mathcal{H}_\alpha^l$ using geometric considerations, which are inspired by Fisher's geometric model of adaptation in Euclidean space [10]. Let individual a be a parent of b, and let $d(a, b) = r$. We consider asexual reproduction as a transition from parent a to a random point b on a sphere $S(a, r)$:

$$b \in S(a, r) := \{\omega : d(a, \omega) = r\}$$

We refer to r as a *radius of mutation*. Suppose that $d(\top,a)=n$ and $d(\top,b)=m$. We define the following probabilities:

$$\begin{aligned} P(r \mid n) &:= P(b \in S(a,r) \mid a \in S(\top,n)),\\ P(m \mid r,n) &:= P(b \in S(\top,m) \mid b \in S(a,r), a \in S(\top,n)),\\ P(r \cap m \mid n) &:= P(b \in S(a,r) \cap S(\top,m) \mid a \in S(\top,n)),\\ P(m \mid n) &:= P(b \in S(\top,m) \mid a \in S(\top,n)). \end{aligned}$$

These probabilities are related as follows:

$$P(m \mid n) = \sum_{r=0}^{l} P(r \cap m \mid n) = \sum_{r=0}^{l} P(m \mid r,n) P(r \mid n). \tag{13}$$

For simple mutation of sequences in $\mathcal{H}_\alpha^l$, the probability that $b \in S(a,r)$ is defined by binomial distribution with probability $\mu \in [0,1]$ of mutation depending on $n = d(\top,a)$:

$$P_\mu(r \mid n) = \binom{l}{r} \mu(n)^r (1-\mu(n))^{l-r}. \tag{14}$$

Probability $P(m \mid r,n)$ is defined by the number of elements in the spheres $S(a,r)$, $S(\top,m)$ and their intersection as follows:

$$P(m \mid r,n) = \frac{|S(\top,m) \cap S(a,r)|_{d(\top,a)=n}}{|S(a,r)|}. \tag{15}$$

The number of sequences in the intersection $S(a,r) \cap S(\top,m)$ with condition $d(\top,a)=n$ is computed by the following formula:

$$|S(\top,m) \cap S(a,r)|_{d(\top,a)=n} = \sum (\alpha-2)^{r_0} \binom{n-r_-}{r_0} (\alpha-1)^{r_+} \binom{l-n}{r_+} \binom{n}{r_-}, \tag{16}$$

where triple summation runs over r_0, r_+ and r_- satisfying conditions $r_+ \in [0, (r+m-n)/2]$, $r_- \in [0, (n-|r-m|)/2]$, $r_- - r_+ = n - \max\{r,m\}$ and $r_0 + r_+ + r_- = \min\{r,m\}$. These conditions can be obtained from metric inequalities for r, m and n (e.g. $|n-m| \le r \le n+m$). The number of sequences in $S(a,r) \subset \mathcal{H}_\alpha^l$ is

$$|S(a,r)| = (\alpha-1)^r \binom{l}{r}. \tag{17}$$

Equations (14)–(17) can be substituted into (13) to obtain the precise expression for transition probability $P_\mu(m \mid n)$ in Hamming space $\mathcal{H}_\alpha^l$.

4 Solutions for Special Cases and Simulation Results

In this section, we derive optimal control functions $\mu(n)$ for several special cases and then evaluate their performance. Given a mutation rate control function $\mu(n)$, we can compute operator $T_{\mu(n)}$ using (13) for transition probabilities $P_\mu(m \mid n)$ in a Hamming space $\mathcal{H}_\alpha^l$. Table 1 lists the expected times of convergence of the resulting processes to the optimal state $x = \sup f(\omega)$, computed by (5) using corresponding absorbing Markov chain. As a reference, Table 1 reports also the expected time for a process with a constant mutation rate $\mu = 1/l$, which corresponds to the error threshold [9, 15] and is sometimes considered optimal (e.g. [16]). Then we use powers $T_{\mu(n)}^t$ of the Markov operators to simulate the processes on a digital computer. The examples of resulting evolutions in time for $\mathcal{H}_2^{10}$ are shown in Fig. 4, and Fig. 5 shows the corresponding evolutions in information.

4.1 Optimal Mutation Rate for Next Generation

Let us consider mutation rate maximizing expected fitness of the next generation. This corresponds to problem (1) with $\lambda = 1$, and it corresponds to minimization of the following conditional expectation:

$$\mathbb{E}_\mu\{m \mid n\} = \sum_{m=0}^{l} m\, P_\mu(m \mid n).$$

Figure 1 shows level sets of $\mathbb{E}_\mu\{m \mid n\}$ as a function of $n = d(\top, a)$ in $\mathcal{H}_2^{30}$ and different mutation rates μ. One can show that mutation rate optimizing the next generation is the following step function:

$$\mu(n) := \begin{cases} 0 & \text{if } n < l(1 - 1/\alpha), \\ \frac{1}{2} & \text{if } n = l(1 - 1/\alpha), \\ 1 & \text{otherwise.} \end{cases}$$

Table 1 Expected times $\mathbb{E}\{t\}$ of convergence to optimum in Hamming spaces $\mathcal{H}_\alpha^l$ using Markov processes for different controls $\mu(n)$ of mutation rate

$\mu(n)$	$\mathcal{H}_2^{10}$	$\mathcal{H}_4^{10}$	$\mathcal{H}_2^{30}$
Constant $1/l$	$16,6 \cdot 10^2$	$163,3 \cdot 10^4$	$170,4 \cdot 10^7$
Step	∞	∞	∞
Linear n/l	$2,5 \cdot 10^2$	$14,9 \cdot 10^4$	$7,6 \cdot 10^7$
$\max_\mu P_\mu(m < n \mid n)$	$3,8 \cdot 10^2$	$19,9 \cdot 10^4$	$17,8 \cdot 10^7$
$P_0(m < n)$	$13,9 \cdot 10^2$	$570,6 \cdot 10^4$	$256,8 \cdot 10^7$

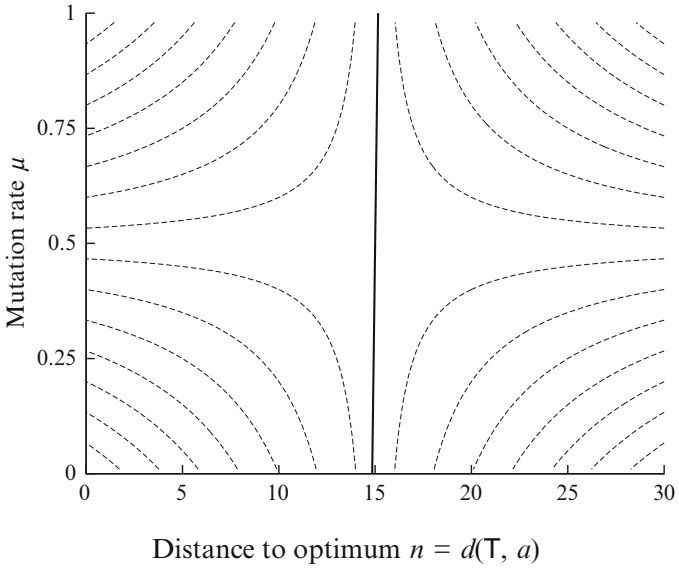

Fig. 1 Expected value $\mathbb{E}_\mu\{m \mid n\}$ of distance $m = d(\top, b)$ to optimum $\top \in \mathcal{H}_2^{30}$ after one transformation $a \mapsto b$ as a function of $n = d(\top, a)$ and mutation rate μ. *Dashed curves* show level sets of $\mathbb{E}_\mu\{m \mid n\}$; *solid curve* shows the minimum

Clearly, the corresponding operator $T_{\mu(n)}$ is not optimal for $t > 1$ generations, because it does not change the distribution of sequences in $\mathcal{H}_\alpha^l$, if $d(\top, \omega) < l(1 - 1/\alpha)$ for all ω. In the space $\mathcal{H}_2^l$ of binary sequences, this occurs after just one generation. Thus, Fig. 4 shows no change in the expected fitness after $t > 1$ for this control function. Figure 5 also shows quite a significant information divergence, so that the optimal information value is not achieved. Note also that for sequences of length $l > 1$, this strategy has infinite expected time to converge to state $m = 0$ (or $x_{s+t} = \sup f(\omega)$) (see Table 1).

4.2 Maximizing Probability of Optimum

Minimization of the convergence time to state $m = 0$ is related to maximization of probability $P_\mu(m = 0 \mid n)$, which has the following expression:

$$P_\mu(m = 0 \mid n) = (\alpha - 1)^{-n} \mu^n (1 - \mu)^{l-n}. \tag{18}$$

Mutation rate μ maximizing this probability is given by taking its derivative to zero:

$$\frac{\mathrm{d}}{\mathrm{d}\mu} P_\mu(m = 0 \mid n) = (\alpha - 1)^{-n} \mu^{n-1} (1 - \mu)^{l-n-1} (n - l\mu) = 0.$$

Together with $\mathrm{d}^2 P_\mu / \mathrm{d}\mu^2 \leq 0$, this gives condition $n - l\mu = 0$ or

$$\mu(n) = \frac{n}{l}. \tag{19}$$

This linear mutation control function has very intuitive interpretation: if sequence a has n letters different from the optimal sequence $\top$, then substitute n letters in the offspring.

One can show that the linear function (19) is optimal for two-valued fitness landscapes with one optimal sequence, such as the Needle in a Haystack discussed in Example 2. This is because expected fitness $\mathbb{E}_{\mu(x)}\{x_{s+t}\}$ in this case is completely defined by probability (18). For other fitness landscapes that are monotonic relative to the Hamming metric, function (19) can be a good approximation of optimal control in terms of (1) with large time constraint λ or (2) with constraint $\upsilon = \sup f(\omega)$. Table 1 shows good convergence times $\mathbb{E}\{t\}$ to the optimum. However, Fig. 4 shows that evolution in time is extremely slow in the initial stage, and in fact not optimal for $t < \mathbb{E}\{t\}$. Figure 5 shows also that performance in terms of information value for this strategy is very poor.

4.3 Maximizing Probability of Success

Consider the following probability:

$$P_\mu(m < n \mid n) = \sum_{m=0}^{n-1} P_\mu(m \mid n).$$

Bäck referred to it as *probability of success* and derived mutation rate $\mu(n)$ maximizing it for the space $\mathcal{H}_2^l$ of binary sequences [2]. Figure 2 shows this curve for $\mathcal{H}_2^{10}$, and similar curves can be obtained for the general case $\mathcal{H}_\alpha^l$ using equations from previous section (Fig. 3).

Although this strategy allows one to achieve good performance, as can be seen from Figs. 4 and 5, it does not solve optimization problems (1) or (3) in general. To see this, observe that maximization of $P_\mu(m < n \mid n)$ is equivalent to maximization of conditional expectation $\mathbb{E}_\mu\{u(m,n) \mid n\} = \sum_m u(m,n) P_\mu(m \mid n)$ of a two-valued utility function

$$u(m,n) = \begin{cases} 1 & \text{if } m < n, \\ 0 & \text{otherwise.} \end{cases}$$

First, this function has only two values, and they depend on two arguments $m = d(\top, b)$ and $n = d(\top, a)$. Thus, u does not correspond to fitness functions with

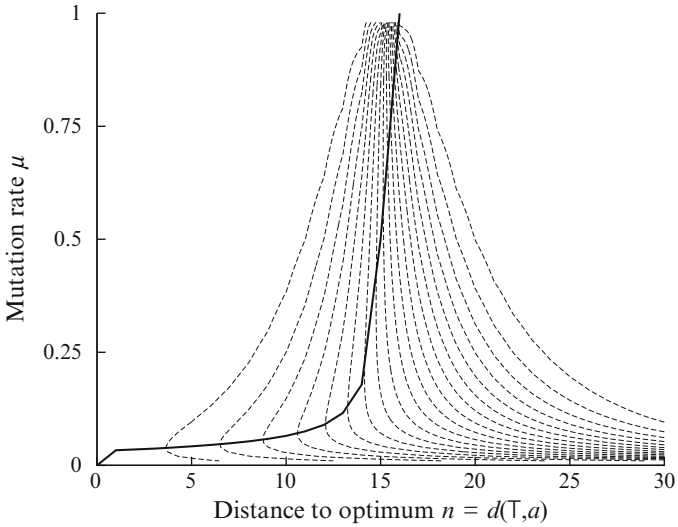

Fig. 2 Probability of "success" $P_\mu(m < n \mid n)$ that b is closer to $\top \in \mathcal{H}_2^{30}$ than a after one transformation $a \mapsto b$ as a function of $n = d(\top, a)$ and mutation rate μ. *Dashed curves* show level sets of $P_\mu(m < n \mid n)$; *solid curve* shows the maximum

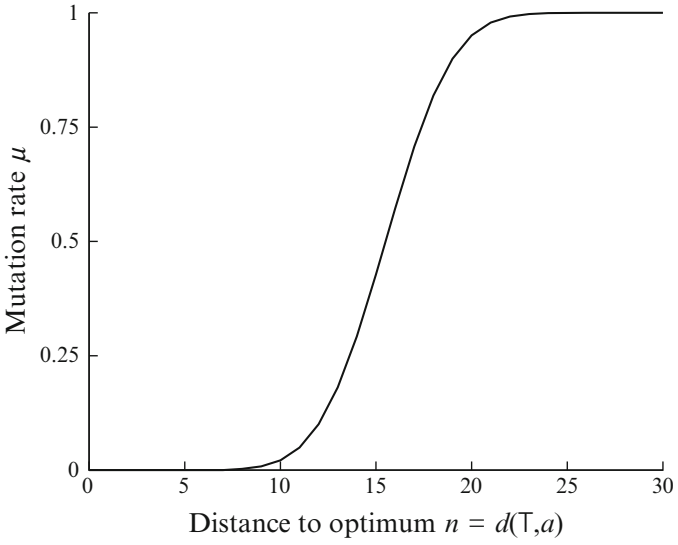

Fig. 3 Probability $P_0(m < n)$, computed as cumulative distribution function of $P_0(m)$ in $\mathcal{H}_2^{30}$, defined by (23)

more than two values, such as the negative distance in Example 1. Note also that fitness usually depends on just one argument (i.e. on the genotype of one individual). Second, the optimization is done for one transition (i.e. next generation), while we

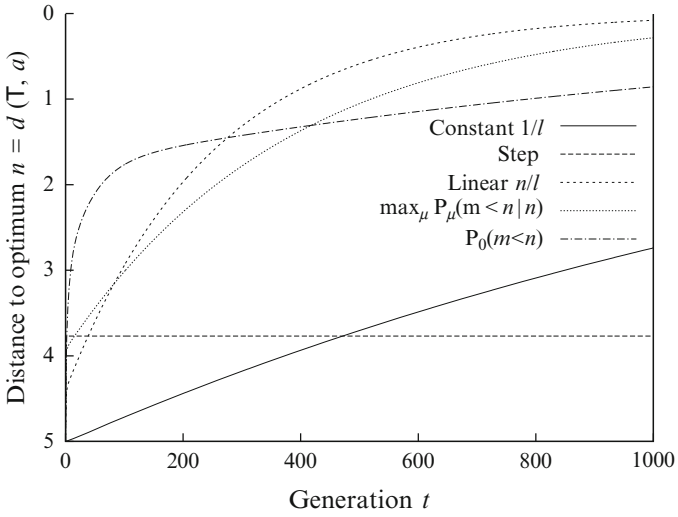

Fig. 4 Expected distance to optimum $\top \in \mathcal{H}_2^{10}$ as a function of generation t (time). Different curves correspond to different controls $\mu(n)$ of mutation rate

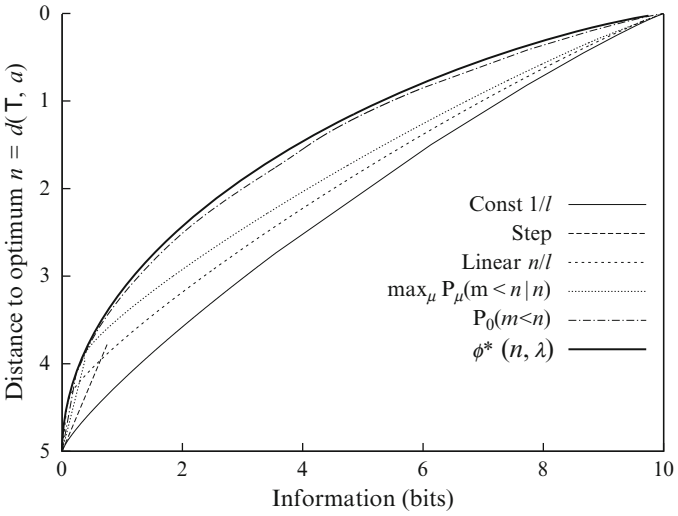

Fig. 5 Expected distance to optimum $\top \in \mathcal{H}_2^{10}$ as a function of information divergence λ from initial distribution. Different curves correspond to different controls $\mu(n)$ of mutation rate; $\overline{\phi}(\lambda)$ represents theoretical optimum

are interested in a mutation rate control maximizing expected fitness after $t > 1$ generations. In fact, one can see from Table 1 that linear control (19) of the mutation rate gives shorter expected times of convergence into absorbing state $m = 0$.

4.4 Minimum Information Rate

Let us consider the problem of controlling a mutation rate to maximize the evolution in information, as defined by the optimal value function $\overline{\phi}(\lambda)$ in (7) or its inverse (8). As stated earlier, the optimal transition kernels for these problems belong to an exponential family (11), and for the transitions in a Hamming space with $f(\omega) = -d(\top, \omega)$, and using notation $n = d(\top, a)$, $m = d(\top, b)$, the transition kernel has the form

$$P_\beta(m \mid n) = \mathrm{e}^{\beta(n-m)-\Psi_{(n-m)}(\beta)}\, P(m). \tag{20}$$

The difference $n - m$ represents fitness value of m relative to n; $\exp\{\Psi_{(n-m)}(\beta)\}$ is the normalizing factor, which depends on β and n. Given an initial distribution $P(n)$, one can obtain its transformation $P(m) = T_\beta P(n)$, where operator T_β is defined by transition probabilities above. Thus, the optimal value function $\overline{\phi}(\lambda)$ can be computed using $\beta \in \mathbb{R}$ as parameter—its argument λ is the information divergence $I_{\mathrm{KL}}(P(m), P(n))$, and its values are the expected fitness $\mathbb{E}\{-m\} = -\sum m\, P(m)$. An example of function $\overline{\phi}(\lambda)$ for $\mathcal{H}_2^{10}$ is shown in Fig. 5. Our task is to define a mutation control function $\mu(n)$ such that the evolution defined by the corresponding Markov operator $T_{\mu(n)}$ achieves optimal values $\overline{\phi}(\lambda)$.

Recall that given random variable $(\Omega, \mathcal{F}, P)$, the value $h(\omega) = -\ln P(\omega)$ is called *random entropy* of outcome ω. In fact, it can be computed as information divergence $I_{\mathrm{KL}}(\delta_\omega, P(\omega)) = -\ln P(\omega)$ of the Dirac measure δ_ω. Entropy is the expected value $\mathbb{E}\{h(\omega)\} = -\sum[\ln P(\omega)]\, P(\omega)$. We can also define random information $\iota(\omega, \xi)$ of two variables as $h(\omega) - h(\omega \mid \xi) = \ln[P(\omega \mid \xi)/P(\omega)]$, and its expected value with respect to joint distribution $P(\omega, \xi)$ is Shannon's mutual information [17]. Conditional probability can be expressed using $\iota(\omega, \xi)$:

$$P(\omega \mid \xi) = \mathrm{e}^{\iota(\omega,\xi)}\, P(\omega).$$

Comparing this to (20), one can see that the quantity $\beta(n-m) - \Psi_{(n-m)}(\beta)$ plays a role of random information $\iota(m, n)$. In fact, one can show that the Legendre–Fenchel dual of $\Psi_f(\beta)$ is the inverse optimal value function $\overline{\phi}^{-1}(\upsilon) = \sup\{\beta\upsilon - \Psi_f(\beta)\}$, and it is defined by (8) as the minimal information subject to $\mathbb{E}\{x\} \geq \upsilon$.

To see how mutation rate $\mu(n)$ can be related to information, let us write transition probability (13) for a Hamming space in the exponential form:

$$P_\mu(m \mid n) = \sum_{r=0}^{l} \mathrm{e}^{r\ln\mu(n)+(l-r)\ln[1-\mu(n)]}\, \frac{|S(a,r) \cap S(\top, m)|_n}{(\alpha-1)^r}. \tag{21}$$

Our experiments show that optimal values $\overline{\phi}(\lambda)$ are achieved if random entropy $h(n) = -\ln\mu(n)$ is identified with $h(m < n \mid n) = -\ln P_0(m < n)$, where

$P_0(m < n)$, shown on Fig. 3, is computed as the cumulative distribution function of the "least informed" distribution $P_0(m)$:

$$\mu(n) = P_0(m < n) = \sum_{m=0}^{n-1} P_0(m). \tag{22}$$

Here, the distribution $P_0(m) := P_0(\omega \in S(\top, m))$ is obtained assuming a uniform distribution $P_0(\omega) = \alpha^{-l}$ of sequences in $\mathcal{H}_\alpha^l$. Thus, $P_0(m)$ can be obtained by counting sequences in the spheres $S(\top, n) \subset \mathcal{H}_\alpha^l$, and it corresponds to binomial distribution with $\mu = 1 - 1/\alpha$:

$$P_0(m) = \binom{l}{m} \mu^m (1-\mu)^{l-m} = \binom{l}{m} \frac{(\alpha-1)^m}{\alpha^l}. \tag{23}$$

In this case, $\mathbb{E}\{m\} = l\mu = l(1 - 1/\alpha)$.

Control of mutation rate by function (22) has the following interpretation: if sequence a has n letters different from the optimal sequence $\top$, then substitute each letter in the offspring with a probability that $d(\top, b) = m < n$. We refer to such control as *minimum information*, because it achieves the same effect as using exponential probability (20) for minimal information $\iota(m, n) = \overline{\phi}^{-1}(\upsilon)$.

Figure 5 shows that this strategy achieves almost perfectly theoretical optimal information value $\overline{\phi}(\lambda)$. Perhaps, even more interesting is that this strategy is optimal in the initial stages of evolution in time, as seen in Fig. 4. Table 1 shows that convergence to the optimal state is very slow. However, Fig. 4 shows that the expected fitness is higher than for any other strategy even after generation $t = 250$, which is the smallest expected convergence time in Table 1. Similar results were observed in other Hamming spaces. Interestingly, the performance of the minimal information strategy in terms of cumulative objective function (3) is also better than other strategies during significant part of the evolution.

5 Discussion

We have considered differences between problems of optimization of evolution in time and optimization of evolution in information. These problems have been studied in relation to optimization of mutation rate in evolutionary algorithms and biological applications. Traditional approach to such problems is based on sequential optimization using methods of dynamic programming and approximate numerical solutions. However, in many practical applications the complexity overwhelms even the most powerful computers. Even in the most simple biological systems, dimensionality of the corresponding spaces of sequences and time horizon make sequential optimization intractable.

On the other hand, optimization of evolution in information can be formulated as convex optimization, and analytical solutions are often available. These solutions define performance bounds against which various algorithms can be evaluated and optimal or nearly optimal solutions can be found. Our results suggest that optimization of evolution in information can also help solve sequential optimization problems. This may provide an alternative way to tackle optimization problems, for which traditional methods have not been effective.

Acknowledgements This work was supported by UK EPSRC grant EP/H031936/1.

References

1. Amari, S.I.: Differential-Geometrical Methods of Statistics. In: Lecture Notes in Statistics, vol. 25. Springer, Berlin (1985)
2. Bäck, T.: Optimal mutation rates in genetic search. In: Forrest, S. (ed.) Proceedings of the 5th International Conference on Genetic Algorithms, pp. 2–8. Morgan Kaufmann (1993)
3. Belavkin, R.V.: Bounds of optimal learning. In: 2009 IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning, pp. 199–204. IEEE, Nashville, TN, USA (2009)
4. Belavkin, R.V.: Information trajectory of optimal learning. In: Hirsch, M.J., Pardalos, P.M., Murphey, R. (eds.) Dynamics of Information Systems: Theory and Applications, Springer Optimization and Its Applications Series, vol. 40. Springer, Berlin (2010)
5. Belavkin, R.V.: On evolution of an information dynamic system and its generating operator. Optimization Letters (2011). DOI:10.1007/s11590-011-0325-z
6. Bellman, R.E.: Dynamic Programming. Princeton University Press, Princeton, NJ (1957)
7. Bernstein, D.S., Hyland, D.C.: The optimal projection/maximum entropy approach to designing low-order, robust controllers for flexible structures. In: Proceedings of 24th Conference on Decision and Control, pp. 745–752. Ft. Lauderdale, FL (1985)
8. Chentsov, N.N.: Statistical Decision Rules and Optimal Inference. Nauka, Moscow, U.S.S.R. (1972). In Russian, English translation: Providence, RI: AMS, 1982
9. Eigen, M., McCaskill, J., Schuster, P.: Molecular quasispecies. J. Phys. Chem. **92**, 6881–6891 (1988)
10. Fisher, R.A.: The Genetical Theory of Natural Selection. Oxford University Press, Oxford (1930)
11. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
12. Kalman, R.E., Bucy, R.S.: New results in linear filtering and prediction theory. Trans. ASME Basic Eng. **83**, 94–107 (1961)
13. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**(1), 79–86 (1951)
14. Nix, A.E., Vose, M.D.: Modeling genetic algorithms with Markov chains. Ann. Math. Artif. Intell. **5**(1), 77–88 (1992)
15. Nowak, M.A.: Evolutionary Dynamics: Exploring the Equations of Life. Harvard University Press, Cambridge (2006)
16. Ochoa, G.: Setting the mutation rate: Scope and limitations of the $1/l$ heuristics. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002), pp. 315–322. Morgan Kaufmann, San Francisco, CA (2002)
17. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 and 623–656 (1948)

18. Stratonovich, R.L.: Optimum nonlinear systems which bring about a separation of a signal with constant parameters from noise. Radiofizika **2**(6), 892–901 (1959)
19. Stratonovich, R.L.: On value of information. Izv. USSR Acad. Sci. Tech. Cybern. **5**, 3–12 (1965) (In Russian)
20. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA (1998)
21. Tsypkin, Y.Z.: Foundations of the Theory of Learning Systems. In: Mathematics in Science and Engineering. Academic, New York (1973)
22. Yanagiya, M.: A simple mutation-dependent genetic algorithm. In: Forrest, S. (ed.) Proceedings of the 5th International Conference on Genetic Algorithms, p. 659. Morgan Kaufmann (1993)

Integration Principle as the Master Equation of the Dynamics of an Information System

Victor Korotkikh and Galina Korotkikh

Abstract In the paper we consider the hierarchical network of prime integer relations as a system of information systems. The hierarchical network is presented by the unity of its two equivalent forms, i.e., arithmetical and geometrical. In the geometrical form a prime integer relation becomes a two-dimensional pattern made of elementary geometrical patterns. Remarkably, a prime integer relation can be seen as an information system itself functioning by the unity of the forms. Namely, while through the causal links of a prime integer relation the information it contains is instantaneously processed and transmitted, the elementary geometrical patterns take the shape to simultaneously reproduce the prime integer relation geometrically. Since the effect of a prime integer relation as an information system is entirely given by the two-dimensional geometrical pattern, the information can be associated with its area. We also consider how the quantum of information of a prime integer relation can be represented by using space and time as dynamical variables. Significantly, the holistic nature of the hierarchical network makes it possible to formulate a single universal objective of a complex system expressed in terms of the integration principle. We suggest the integration principle as the master equation of the dynamics of an information system in the hierarchical network.

Keywords Information system • Prime integer relation • Quantum of information • Complexity • Integration principle

V. Korotkikh (✉) • G. Korotkikh
School of Information and Communication Technology CQUniversity,
Mackay, QLD 4740, Australia
e-mail: v.korotkikh@cqu.edu.au; g.korotkikh@cqu.edu.au

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*,
Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_2,

1 Introduction

In the paper we consider the hierarchical network of prime integer relations as a system of information systems.

For this purpose in Sect. 2 we present the hierarchical network by the unity of its two equivalent forms, i.e., arithmetical and geometrical. In particular, we discuss that in the geometrical form a prime integer relation becomes a two-dimensional pattern made of elementary geometrical patterns.

Remarkably, a prime integer relation can be seen as an information system functioning by the unity of two forms. Namely, while through the causal links of a prime integer relation the information is instantaneously processed and transmitted for the prime integer relation to be defined, the elementary geometrical patterns take the shape to simultaneously reproduce the prime integer relation geometrically. Therefore, a prime integer relation has a very important property to process and transmit information to the parts so that they can operate together for the system to exist and function as a whole.

Since the effect of a prime integer relation as an information system is entirely given by the two-dimensional geometrical pattern, the information can be associated with the geometrical pattern and its area in particular. This suggests that in a prime integer relation the information is made of quanta given by the elementary geometrical patterns and measured by their areas.

In Sect. 3 we consider how the quantum of information of a prime integer relation can be represented by using space and time as dynamical variables.

In Sect. 4 we discuss that the holistic nature of the hierarchical network makes it possible to formulate a single universal objective of a complex system expressed in terms of the integration principle. We suggest the integration principle as the master equation of the dynamics of an information system in the hierarchical network.

2 The Hierarchical Network of Prime Integer Relations as a System of Information Systems

The hierarchical network has been defined within the description of complex systems in terms of self-organization processes of prime integer relations [1–7]. Remarkably, in the hierarchical network arithmetic and geometry are unified by two equivalent forms, i.e., arithmetical and geometrical. At the same time, the arithmetical and geometrical forms play the different roles. For example, while the arithmetical form sets the relationships between the parts of a system, the geometrical form makes it possible to measure the effect of the relationships on the parts.

In the arithmetical form the hierarchical network comes into existence by the totality of the self-organization processes of prime integer relations. Starting with

the integers the processes build the hierarchical network under the control of arithmetic as one harmonious and interconnected whole, where not even a minor change can be made to any of its elements.

In the description a complex system is defined by a number of global quantities conserved under self-organization processes. The processes build hierarchical structures of prime integer relations, which determine the system. Importantly, since a prime integer relation expresses a law between the integers, the complex system becomes governed by the laws of arithmetic realized by the self-organization processes of prime integer relations.

Remarkably, a prime integer relation of any level can be considered as a complex system itself. Indeed, it is formed by a process from integers as the initial building blocks and then from prime integer relations of the levels below with the relationships set by arithmetic. Because each and every element in the formation is necessary and sufficient for the prime integer relation to exist, we call such an integer relation prime.

In the geometrical form the formation of a prime integer relation can be isomorphically represented by the formation of two-dimensional geometrical patterns [1–3]. In particular, in the geometrical form a prime integer relation, as well as a corresponding law of arithmetic, becomes expressed by a two-dimensional geometrical pattern made of elementary geometrical patterns, i.e., the quanta of the prime integer relation. Notably, when the areas of the elementary geometrical patterns are calculated they turn out to be quantized [8–10].

Due to the isomorphism of the forms, the relationships in a prime integer relation determine the shape of the elementary patterns to make the whole geometrical pattern. Strictly controlled by arithmetic, the shapes of the elementary geometrical patterns cannot be changed even a bit without breaking the relationships and thus the prime integer relation.

Significantly, a prime integer relation can be seen as an information system functioning through the unity of the forms. In particular, while through the causal links of a prime integer relation the information it contains is instantaneously processed and transmitted for the prime integer to become defined, the elementary geometrical patterns take the shape to simultaneously reproduce the prime integer relation geometrically.

Since the effect of a prime integer relation as an information system is entirely given by the two-dimensional geometrical pattern, the information can be associated with the geometrical pattern and its area in particular. This suggests that in a prime integer relation the information is made of quanta given by the elementary geometrical patterns and measured by their areas [10].

As a result, a concept of information based on the self-organization processes of prime integer relations and thus arithmetic can be defined.

Now let us illustrate the general results. It has been shown that if under the transition from one state $s = s_1 \dots s_N$ to another state $s' = s'_1 \dots s'_N$ at level 0 $k \geq 1$ quantities of the complex system remain invariant, then k Diophantine equations

$$(m+N)^{k-1}\Delta s_1 + (m+N-1)^{k-1}\Delta s_2 + \cdots + (m+1)^{k-1}\Delta s_N = 0,$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$(m+N)^{1}\Delta s_1 + (m+N-1)^{1}\Delta s_2 + \cdots + (m+1)^{1}\Delta s_N = 0,$$

$$(m+N)^{0}\Delta s_1 + (m+N-1)^{0}\Delta s_2 + \cdots + (m+1)^{0}\Delta s_N = 0 \qquad (1)$$

and an inequality

$$(m+N)^{k}\Delta s_1 + (m+N-1)^{k}\Delta s_2 + \ldots + (m+1)^{k}\Delta s_N \neq 0 \qquad (2)$$

take place [1–3].

In particular, it is assumed that in the state $s = s_1 \ldots s_N$ there are $|s_i|$ of integers $m+N-i+1,\ i = 1, \ldots, N$ "charged" positively, if $s_i > 0$, or "charged" negatively, if $s_i < 0$. Similarly, in the state $s' = s'_1 \ldots s'_N$ there are $|s'_i|$ of integers $m+N-i+1,\ i = 1, \ldots, N$ "charged" positively, if $s'_i > 0$, or "charged" negatively, if $s'_i < 0$. At the same time m and $N,\ N \geq 2$ are integers and

$$\Delta s_i = s'_i - s_i,\ i = 1, \ldots, N,$$

where $s_i, s'_i \in I$ and I is a set of integers.

Notably, integers

$$m+N, m+N-1, \ldots, m+1$$

appear as the initial building blocks of the system and to make the transition from the state $s = s_1 \ldots s_N$ to the state $s' = s'_1 \ldots s'_N$ it is required that $|\Delta s_i|$ of integers

$$m+N-i+1,\ i = 1, \ldots, N$$

have to be generated from the "vacuum" positively "charged," if $\Delta s_i > 0$, or "charged" negatively, if $\Delta s_i < 0$.

Let us consider the Diophantine equations (1) when the PTM (Prouhet-Thue-Morse) sequence of length N

$$\eta = +1-1-1+1-1+1+1-1\ldots = \eta_1 \ldots \eta_N$$

specifies a solution

$$\Delta s_i = \eta_i,\ i = 1, \ldots, N$$

for $N = 2^k,\ k = 1, 2, \ldots$.

Namely, in this case the Diophantine equations (1) and inequality (2) become

$$N^{k-1}\eta_1 + (N-1)^{k-1}\eta_2 + \cdots + 1^{k-1}\eta_N = 0,$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot$$

$$N^{1}\eta_1 + (N-1)^{1}\eta_2 + \cdots + 1^{1}\eta_N = 0,$$

$$N^{0}\eta_1 + (N-1)^{0}\eta_2 + \cdots + 1^{0}\eta_N = 0 \qquad (3)$$

and

$$N^k\eta_1 + (N-1)^k\eta_2 + \cdots + 1^k\eta_N \neq 0, \tag{4}$$

where $m = 0$.

For example, when $N = 16$ we can explicitly write (3) and (4) as

$$\begin{gathered}
+16^3 - 15^3 - 14^3 + 13^3 - 12^3 + 11^3 + 10^3 - 9^3 \\
-8^3 + 7^3 + 6^3 - 5^3 + 4^3 - 3^3 - 2^3 + 1^3 = 0 \\
+16^2 - 15^2 - 14^2 + 13^2 - 12^2 + 11^2 + 10^2 - 9^2 \\
-8^2 + 7^2 + 6^2 - 5^2 + 4^2 - 3^2 - 2^2 + 1^2 = 0 \\
+16^1 - 15^1 - 14^1 + 13^1 - 12^1 + 11^1 + 10^1 - 9^1 \\
-8^1 + 7^1 + 6^1 - 5^1 + 4^1 - 3^1 - 2^1 + 1^1 = 0 \\
+16^0 - 15^0 - 14^0 + 13^0 - 12^0 + 11^0 + 10^0 - 9^0 \\
-8^0 + 7^0 + 6^0 - 5^0 + 4^0 - 3^0 - 2^0 + 1^0 = 0
\end{gathered} \tag{5}$$

and

$$\begin{gathered}
+16^4 - 15^4 - 14^4 + 13^4 - 12^4 + 11^4 + 10^4 - 9^4 \\
-8^4 + 7^4 + 6^4 - 5^4 + 4^4 - 3^4 - 2^4 + 1^4 \neq 0.
\end{gathered} \tag{6}$$

Next we consider one of the self-organization processes of prime integer relations that can be associated with the system of integer relations (5) and inequality (6).

The self-organization process starts as integers $16, \ldots, 1$ are generated from the "vacuum" to appear at level 0 positively or negatively "charged" depending on the sign of the corresponding element in the PTM sequence. Then the integers combine into pairs and make up the prime integer relations of level 1. Following a single organizing principle [1–3] the process continues as long as arithmetic allows the prime integer relations of a level to form the prime integer relations of the higher level (Fig. 1).

In the geometrical form, which is specified by two parameters $\varepsilon \geq 1$ and $\delta \geq 1$, the self-organization process become isomorphically represented by transformations of two-dimensional patterns (Fig. 2). Remarkably, under the isomorphism a prime integer relation turns into a corresponding geometrical pattern, which can be viewed as the prime integer relation itself, but only expressed geometrically.

At level 0 the geometrical pattern of integer $16 - i + 1,\ i = 1, \ldots, 16$ is given by the region enclosed by the boundary curve, i.e., the graph of the function

$$\Psi_1^{[0]}(t) = \eta_i\delta,\ t_{i-1} \leq t < t_i,$$

the vertical lines $t = t_{i-1},\ t = t_i$ and the t-axis, where $t_j = j\varepsilon,\ j = 0, \ldots, 16$.

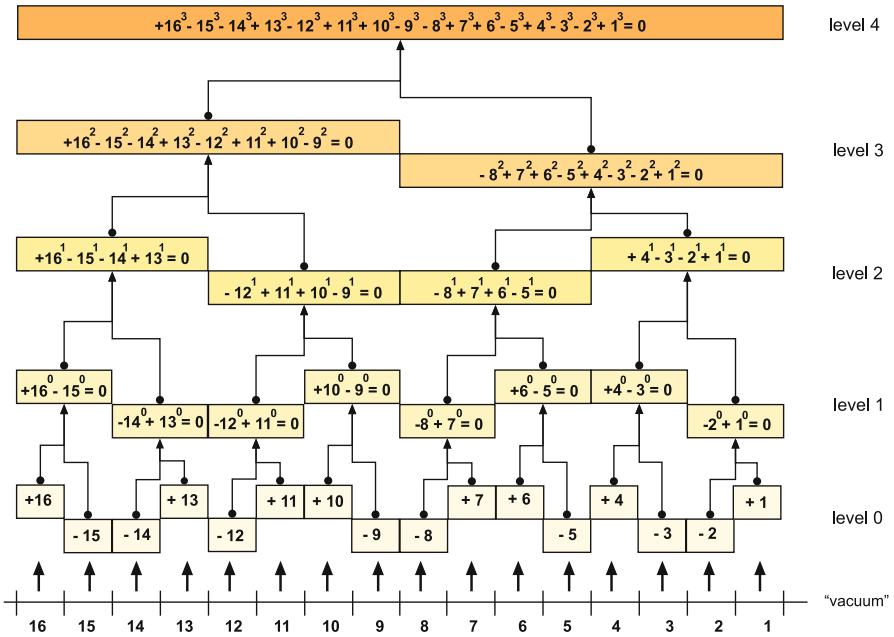

Fig. 1 The hierarchical structure of prime integer relations built by the process

At level $l = 1, 2, 3, 4$ the geometrical pattern of the ith $i = 1, \ldots, 2^{4-l}$ prime integer relation is defined by the region enclosed by the boundary curve, i.e., the graph of the function

$$\Psi_1^{[l]}(t),\ t_{2^l(i-1)} \le t \le t_{2^l i},$$

and the t-axis.

As the integers of level $l = 0$ or the prime integer relations of level $l = 1, 2, 3$ form the prime integer relations of level $l + 1$, under the integration of the function

$$\Psi_1^{[l]}(t),\ t_0 \le t \le t_{16}$$

subject to

$$\Psi_1^{[l+1]}(t_0) = 0,$$

the geometrical patterns of level l transform into the geometrical patterns of level $l + 1$.

Remarkably, the geometrical pattern of a prime integer relation is composed of elementary geometrical patterns, i.e., the quanta of the prime integer relation.

For example, the ith $i = 1, \ldots, 8$ elementary geometrical pattern of the prime integer relation

$$+16^2 - 15^2 - 14^2 + 13^2 - 12^2 + 11^2 + 10^2 - 9^2 = 0$$

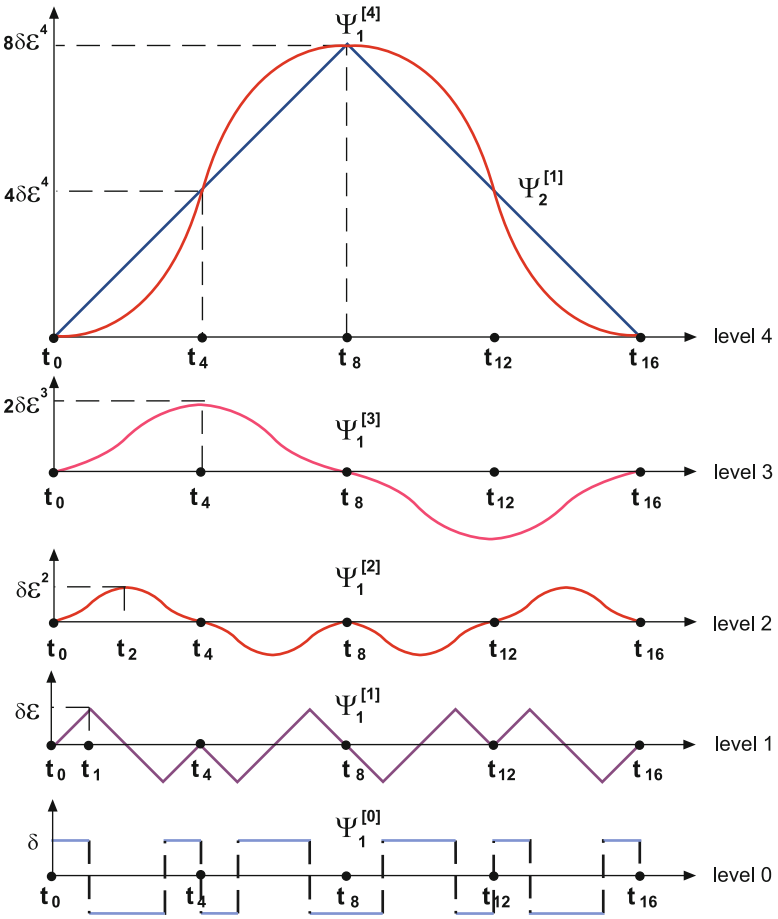

Fig. 2 The hierarchical structure of geometrical patterns

is the region enclosed by the boundary curve, i.e., the graph of the function

$$\Psi_1^{[3]}(t),\ t_{i-1} \le t \le t_i,$$

the vertical lines $t = t_{i-1},\ t = t_i$ and the t-axis.

Significantly, the areas of the elementary geometrical patterns of a prime integer relation turn out to be quantized. For instance, the areas of the elementary geometrical patterns $G_{14}, \ldots, G_{16,4}$ of the prime integer relation

$$+16^3 - 15^3 - 14^3 + 13^3 - 12^3 + 11^3 + 10^3 - 9^3$$
$$-8^3 + 7^3 + 6^3 - 5^3 + 4^3 - 3^3 - 2^3 + 1^3 = 0$$

produce a discrete spectrum of quantized values

$$A(G_{14}), \ldots, A(G_{16,4}) = \frac{1}{120}, \frac{29}{120}, \frac{149}{120}, \frac{361}{120}, \frac{599}{120}, \frac{811}{120}, \frac{931}{120}, \frac{959}{120}, \frac{959}{120}, \frac{931}{120}, \frac{811}{120}, \frac{599}{120}, \frac{361}{120}, \frac{149}{120}, \frac{29}{120}, \frac{1}{120},$$

when $\varepsilon = 1$ and $\delta = 1$ [9, 10].

Notably, the area $A(G_{i4})$ of an elementary geometrical pattern G_{i4}, $i = 1, \ldots, 16$ can be given by the equation

$$A(G_{i4}) = h\nu(G_{i4}),$$

where

$$h = \frac{1}{120}$$

and $\nu(G_{i4})$ is a corresponding number.

When the prime integer relation becomes defined, the amount of information $I(G_{i4})$ processed and transmitted to the ith elementary geometrical pattern G_{i4}, $i = 1, \ldots, 16$ is given by the area of the geometrical pattern

$$I(G_{i4}) = A(G_{i4}).$$

Now, let us illustrate the processing and transmission of information by using a prime integer relation

$$2^1 \Delta s_1 + 1^1 \Delta s_2 + 0^1 \Delta s_3 = 0 \tag{7}$$

of level 2, where $\Delta s_1 = +1$, $\Delta s_2 = -2$, and $\Delta s_3 = +1$. The prime integer relation (7) is formed from a prime integer relation

$$2^0 \Delta s_1 + 1^0 \Delta s_2 + 0^0 \Delta s_3 = 0 \tag{8}$$

of level 1. The integer relation (8) is prime by definition, because all integers, i.e., one positively "charged" integer 2, two negatively "charged" integers 1, as one indivisible block, and one positively "charged" integer 0, are necessary and sufficient for the formation of the prime integer relation.

Next we consider an integer relation

$$2^1 \Delta s_1 + 1^1 \Delta s_2 = 0, \tag{9}$$

where, in comparison with (7), the term $0^1 \Delta s_3$ is hidden. We can rewrite (9) as

$$2^1 \Delta s_1 = -1^1 \Delta s_2. \tag{10}$$

Although the integer relation (9) simplifies things, yet in our illustration it gives an interesting interpretation of the equals sign.

In particular, as soon as the integer relation (9) becomes operational by setting $\Delta s_2 = -2$, we can see from (10) that the information is instantaneously processed and through the equals sign, working and looking like a channel, transmitted for Δs_1 to be set $\Delta s_1 = 1$, so that the parts can simultaneously give rise to the integer relation

$$2^1 \times 1 + 1^1 \times (-2) = 0$$

emerging as one whole.

Therefore, a prime integer relation, as an information system, has a very important property. Namely, a prime integer relation has the power to process and transmit information to the parts, so that they can operate together for the system to exist and function as a whole. Remarkably, this property of the prime integer relation can be expressed in terms of space and time as dynamical variables [8–10].

A quantum of a prime integer relation, as a quantum of information, is given by an elementary geometrical pattern fully defined by the boundary curve and the area. Therefore, the representation of the quantum of information can be done by the representation of the boundary curve and the area of the geometrical pattern. For this purpose an elementary part could come into existence.

In particular, once the boundary curve is specified by the space and time variables of the elementary part and the area associated with its energy, the quantum of information becomes represented by the elementary part. As a result, the law of motion of the elementary part is determined by the law of arithmetic the prime integer relation realizes.

Significantly, the area of the geometrical pattern of a prime integer relation can be conserved under a renormalization. Therefore, the energy becomes an important variable of the representation [8–10].

For example, in Fig. 2 the renormalization is illustrated by a function

$$\Psi_2^{[1]}(t),\ t_0 \le t \le t_{16}.$$

Notably, the area of the geometrical pattern of the prime integer relation

$$+16^3 - 15^3 - 14^3 + 13^3 - 12^3 + 11^3 + 10^3 - 9^3$$
$$-8^3 + 7^3 + 6^3 - 5^3 + 4^3 - 3^3 - 2^3 + 1^3 = 0$$

remains the same under the renormalization

$$\int_{t_0}^{t_{16}} \Psi_1^{[4]}(t)\mathrm{d}t = \int_{t_0}^{t_{16}} \Psi_2^{[1]}(t)\mathrm{d}t$$

and thus the energy of the elementary parts representing the prime integer relation by their space and time variables is conserved.

3 Representation of the Quantum of Information by Space and Time as Dynamic Variables

Now let us consider how the quantum of information of a prime integer relation can be represented by using space and time as dynamical variables [8–10].

Figure 1 shows that in the arithmetical form there are no relationships between the integers at level 0. On the other side, in the geometrical form (Fig. 2) the boundary curve of the geometrical pattern of integer $16 - i + 1,\ i = 1, \ldots, 16$ is given by the piecewise constant function

$$\Psi_1^{[0]}(t),\ t_{i-1} \le t < t_i$$

and can be represented by the space X_{i0} and T_{i0} time variables of an elementary part P_{i0}.

Namely, as the elementary part P_{i0} makes transition from one state into another at the moment $T_{i0}(t_{i-1}) = 0$ of its local time the space variable $X_{i0}(t_{i-1})$ of the elementary part P_{i0} changes by

$$\Delta X_{i0} = \Psi_1^{[0]}(t_{i-1}) = \eta_i \delta$$

and then stays as it is, while the time variable

$$T_{i0}(t),\ t_{i-1} \le t < t_i,$$

changes independently as the length of the boundary curve

$$\Delta T_{i0}(t) = T_{i0}(t) - T_{i0}(t_{i-1}) = T_{i0}(t) = \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}\Psi_1^{[0]}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t',$$

where

$$\Delta T_{i0} = \lim_{t \to t_i} \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}\Psi_1^{[0]}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t' = \varepsilon.$$

Under the integration of the function

$$\Psi_1^{[l]}(t),\ l = 0, 1, 2, 3,\ t_0 \le t \le t_{16},$$

subject to

$$\Psi_1^{[l+1]}(t_0) = 0,$$

the geometrical patterns of the integers of level $l = 0$ and the prime integer relations of level $l = 1, 2, 3$ transform into the geometrical patterns of the prime integer

relations of level $l+1$. As a result, the boundary curve of an elementary geometrical pattern $G_{il},\ i = 1,\ldots,16$, i.e., the graph of the function

$$\Psi_1^{[l]}(t),\ t_{i-1} \le t \le t_i,$$

transforms into the boundary curve of an elementary geometrical pattern $G_{i,l+1}$, i.e., the graph of the function

$$\Psi_1^{[l+1]}(t),\ t_{i-1} \le t \le t_i.$$

Defined at levels $1, 2, 3, 4$ elementary parts represent the boundary curves of the geometrical patterns by their space and time variables [8–10].

In particular, at level 1 the space variable $X_{i1}(t)$ and the time variable $T_{i1}(t),\ t_{i-1} \le t \le t_i$ of an elementary part $P_{i1}, i = 1,\ldots,16$ become linearly dependent and characterize the motion of the elementary part P_{i1} by

$$\Delta T_{i1}(t)\sin\alpha_i = \Delta X_{i1}(t), \tag{11}$$

where

$$\Delta X_{i1}(t) = X_{i1}(t) - X_{i1}(t_{i-1}) = \Psi_1^{[1]}(t) - \Psi_1^{[1]}(t_{i-1}),$$

$$\Delta T_{i1}(t) = \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}\Psi_1^{[1]}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t' = \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}X_{i1}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t'$$

and the angle α_i is given by

$$\tan\alpha_i = \Psi_1^{[0]}(t_{i-1}).$$

Let

$$\Delta X_{i1} = X_{i1}(t_i) - X_{i1}(t_{i-1})$$

and, since $T_{i1}(t_{i-1}) = 0$,

$$\Delta T_{i1} = T_{i1}(t_i) - T_{i1}(t_{i-1}) = T_{i1}(t_i).$$

The velocity $V_{i1}(t),\ t_{i-1} \le t \le t_i$ of the elementary part P_{i1}, as a dimensionless quantity, can be defined by

$$V_{i1}(t) = \frac{\Delta X_{i1}(t)}{\Delta T_{i1}(t)}. \tag{12}$$

Using (11) and (12), we obtain

$$V_{i1}(t) = \sin\alpha_i$$

and, since the angle α_i is constant, the velocity $V_{i1}(t)$ must also stay constant

$$V_{i1}(t) = V_{i1}.$$

By definition $-1 \leq \sin\alpha_i \leq 1$, so we have

$$-1 \leq V_{i1} \leq 1. \tag{13}$$

Since the velocity V_{i1} is a dimensionless quantity, the condition (13) determines a velocity limit c [9, 10]. Therefore, the dimensional velocity v_{i1} of the elementary part P_{i1} can be given by

$$V_{i1} = \sin\alpha_i = \frac{v_{i1}}{c} \tag{14}$$

and thus $|v_{i1}| \leq c$.

Now let us consider how the times ΔT_{i0} and ΔT_{i1} of the elementary parts P_{i0} and $P_{i1}, i = 1, \ldots, 16$ are connected. From Fig. 2 we can find that

$$\Delta T_{i1} |\cos\alpha_i| = \Delta T_{i0}$$

and, by using (14), we get

$$\Delta T_{i1} = \frac{\Delta T_{i0}}{\sqrt{1 - \frac{v_{i1}^2}{c^2}}}. \tag{15}$$

Since the motions of the elementary parts P_{i0} and P_{i1} have to be realized simultaneously, then, according to (15), the time $T_{i1}(t)$ of the elementary part P_{i1} runs faster than the time $T_{i0}(t)$, $t_{i-1} \leq t \leq t_i$ of the elementary part P_{i0}.

Remarkably, (15) symbolically reproduces the well-known formula connecting the elapsed times in the moving and the stationary systems [11] and allows its interpretation. In particular, as long as one tick of the clock of the moving elementary part P_{i1} takes longer $\Delta T_{i1} > \Delta T_{i0}$ than one tick of the clock of the stationary elementary part P_{i0}, then the time in the moving system will be less than the time in the stationary system.

Notably, at level 1 the motion of the elementary part P_{i1} has the invariant

$$\Delta T_{i1}^2 - \Delta X_{i1}^2 = \varepsilon^2, \tag{16}$$

where features of the Lorentz invariant can be recognized.

Significantly, in the representation of the boundary curve the space and time variables of an elementary part $P_{il}, i = 1, \ldots, 16$ at level $l = 2, 3, 4$ become interdependent. As a result, the boundary curve can be seen as their joint entity defining the local spacetime of the elementary part P_{il}. For the sake of consistency, we consider that the boundary curves at level $l = 0, 1$ also define the local spacetimes of the elementary parts.

In particular, in the representation of the boundary curve given by the graph of the function

$$\Psi_1^{[l]}(t),\ t_{i-1} \le t \le t_i,\ i = 1, \ldots, 16,\ l = 2, 3, 4$$

the space variable $X_{il}(t)$ of the elementary part P_{il} is defined by

$$X_{il}(t) = \Psi_1^{[l]}(t),\ t_{i-1} \le t \le t_i. \tag{17}$$

In its turn the time variable $T_{il}(t)$ of the elementary part P_{il} is defined by the length of the curve

$$\begin{aligned} T_{il}(t) &= \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}\Psi_1^{[l]}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t' \\ &= \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}X_{il}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t',\ t_{i-1} \le t \le t_i. \end{aligned} \tag{18}$$

As a result of (18) and the character of the function $\Psi_1^{[l]}(t)$, the space $X_{il}(t)$ and time $T_{il}(t)$ variables become interdependent [8–10].

Moreover, the motion of the elementary part P_{il} can be defined by the change of the space variable $X_{il}(t)$ with respect to the time variable $T_{il}(t)$. Namely, as the time variable $T_{il}(t)$ changes by

$$\begin{aligned} \Delta T_{il}(t) &= \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}\Psi_1^{[l]}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t' \\ &= \int_{t_{i-1}}^{t} \sqrt{1 + \left(\frac{\mathrm{d}X_{il}(t')}{\mathrm{d}t'}\right)^2}\,\mathrm{d}t', \end{aligned}$$

the space variable $X_{il}(t)$ changes by

$$\Delta X_{il}(t) = \Psi_1^{[l]}(t) - \Psi_1^{[l]}(t_{i-1}).$$

By using (18), we can find that the motion of an elementary part $P_{il},\ i = 1, \ldots, 16,\ l = 2, 3, 4$ has the following invariant

$$\left(\frac{\mathrm{d}T_{il}(t)}{\mathrm{d}t}\right)^2 - \left(\frac{\mathrm{d}X_{il}(t)}{\mathrm{d}t}\right)^2 = 1,$$

while the invariant (16) can be seen as its special case.

Therefore, we have considered how the quanta of information of the prime integer relations can be represented by the local spacetimes of elementary parts.

Figure 2 helps us to understand the resulting structure of the local spacetimes and illustrates how the simultaneous realization of the prime integer relations, as a solution to the Diophantine equations (3), becomes expressed by using space and time variables. Namely, as the prime integer relations turn to be operational, then in the representation of the quanta of information of the prime integer relations the elementary parts of all levels become instantaneously connected and move simultaneously, so that their local spacetimes can reproduce the prime integer relations geometrically.

Thus, the self-organization process of prime integer relations can define a complex information system whose representation in space and time determines the dynamics of the parts preserving the system as a whole.

4 Integration Principle as the Master Equation of the Dynamics of an Information System

The holistic nature of the hierarchical network allows us to formulate a single universal objective of a complex system expressed in terms of the integration principle [12–16]:

In the hierarchical network of prime integer relations a complex system has to become an integrated part of the corresponding processes or the larger complex system.

Significantly, the integration principle determines the general objective of the optimization of a complex system in the hierarchical network.

In the realization of the integration principle the geometrical form of the description can play a special role. In particular, the position of a system in the corresponding processes can be associated with a certain two-dimensional shape, which the geometrical pattern of the optimized system has to take precisely to satisfy the integration principle. Therefore, in the realization of the integration principle it is important to compare the current geometrical pattern of the system with the one required for the system by the integration principle. Since the geometrical patterns are two-dimensional, the difference between their areas can be used to estimate the result.

Moreover, the fact that in the hierarchical network processes progress level by level in one and the same direction and, as a result, make a system more and more complex, suggests a possible way for the efficient realization of the integration principle. Namely, as the complexity of a system increases level by level, the area of its geometrical pattern may monotonically become larger and larger. Consequently, with each next level $l < k$ the geometrical pattern of the system would fit better into

the geometrical pattern specified by the integration principle at level k and deviate more after. In its turn, the performance of the optimized system could increase to attain the global maximum at level $l = k$.

Therefore, the performance of the system might behave as a concave function of the complexity with the global maximum at level k specified by the integration principle.

Extensive computational experiments have been successfully conducted to test the prediction. Moreover, the experiments not only support the claim, but also suggest that the integration principle of a complex system could be efficiently realized in general [12, 13].

Let us consider the integration principle in the context of optimization of NP-hard problems. For this purpose an algorithm $\mathcal{A}$, as a complex system of n computational agents, has been used to minimize the average distance in the travelling salesman problem (TSP).

In the algorithm all agents start in the same city and choose the next city at random. Then at each step an agent visits the next city by using one of the two strategies: random or greedy. In the solution of a problem with N cities the state of the agents at step $j = 1, \ldots, N-1$ can be specified by a binary sequence $s_{1j} \ldots s_{nj}$, where $s_{ij} = +1$, if agent $i = 1, \ldots, n$ uses the random strategy and $s_{ij} = -1$, if the agent uses the greedy strategy, i.e., the strategy to visit the closest city.

The dynamics of the system is realized by the strategies the agents choose step by step and can be encoded by the strategy matrix

$$S = \{s_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, N-1\}.$$

In the experiments the complexity of the algorithm has been tried to be changed monotonically by forcing the system to make the transition from regular behavior to chaos by period doubling. To control the system in this transition a parameter $\vartheta,\ 0 \leq \vartheta \leq 1$ has been introduced. It specifies a threshold point dividing the interval of current distances travelled by the agents into two parts, i.e., successful and unsuccessful. This information is required for an optimal if-then rule [17] each agent uses to choose the next strategy. The rule relies on the PTM sequence and has the following description:

1. If the last strategy is successful, continue with the same strategy.
2. If the last strategy is unsuccessful, consult PTM generator which strategy to use next.

Remarkably, it has been found that for any problem p from a class $\mathcal{P}$ the performance of the algorithm behaves as a concave function of the control parameter with the global maximum at $\vartheta^*(p)$. The global maximums $\{\vartheta^*(p),\ p \in \mathcal{P}\}$ have been then probed to find out whether the complexities of the algorithm and the problem are related.

For this purpose the strategy matrices $\{S(\vartheta^*(p)),\ p \in \mathcal{P}\}$ corresponding to the global maximums $\{\vartheta^*(p),\ p \in \mathcal{P}\}$ to characterize the geometrical pattern of the algorithm and its complexity have been tried.

In particular, the area of the geometrical pattern and the complexity $C(\mathcal{A}(p))$ of the algorithm $\mathcal{A}$ are approximated by the quadratic trace

$$C(\mathcal{A}(p)) = \frac{1}{n^2} tr(V^2(\vartheta^*(p))) = \frac{1}{n^2} \sum_{i=1}^{n} \lambda_i^2$$

of the variance–covariance matrix $V(\vartheta^*(p))$ obtained from the strategy matrix $S(\vartheta^*(p))$, where $\lambda_i,\ i = 1, \ldots, n$ are the eigenvalues of $V(\vartheta^*(p))$.

On the other side, the area of the geometrical pattern and the complexity $C(p)$ of the problem p are approximated by the quadratic trace

$$C(p) = \frac{1}{N^2} tr(M^2(p)) = \frac{1}{N^2} \sum_{i=1}^{N} \lambda_i'^2$$

of the normalized distance matrix

$$M(p) = \{d_{ij}/d_{\max},\ i, j = 1, \ldots, N\},$$

where $\lambda_i',\ i = 1, \ldots, N$ are the eigenvalues of $M(p)$, d_{ij} is the distance between cities i and j and $d_{\max}$ is the maximum of the distances.

To reveal a possible connection between the complexities the points with the coordinates

$$\{x = C(p),\ y = C(\mathcal{A}(p)),\ p \in \mathcal{P}\}$$

have been considered. Remarkably, the result indicates a linear dependence between the complexities and suggests the following optimality condition of the algorithm [13].

If the algorithm $\mathcal{A}$ demonstrates the optimal performance for a problem p, then the complexity $C(\mathcal{A}(p))$ of the algorithm is in the linear relationship

$$C(\mathcal{A}(p)) = 0.67C(p) + 0.33$$

with the complexity $C(p)$ of the problem p.

According to the optimality condition, if the optimal performance takes place, then the complexity of the algorithm has to be in a certain linear relationship with the complexity of the problem.

The optimality condition can be a practical tool. Indeed, for a given problem p, by using the normalized distance matrix $M(p)$, we can calculate the complexity $C(p)$ of the problem p and from the optimality condition find the complexity $C(\mathcal{A}(p))$ of the algorithm $\mathcal{A}$. Then, to obtain the optimal performance of the algorithm $\mathcal{A}$ for the problem p, we only need to adjust the control parameter ϑ for the algorithm to work with the required complexity.

Since the geometrical pattern of a system is used to define the complexity of the system, the optimality condition may be interpreted in terms of the integration principle. Namely, when the algorithm shows the optimal performance

for a problem, the geometrical pattern of the algorithm may fit exactly into the geometrical pattern of the problem. Therefore, the algorithm, as a complex system, may become an integrated part of the processes characterizing the problem.

Now let us discuss the computational results in the context of the development of efficient quantum algorithms.

The main idea of quantum algorithms is to make use of quantum entanglement, which, as a physical phenomenon, has not been well understood so far. Moreover, the sensitivity of quantum entanglement is not technologically tamed to support the computations [18]. Conceptually, in a quantum TSP algorithm the wave function has to be evolved to maximize the probability of the shortest routes to be measured. However, it is still unknown how to run the evolution in order to make a quantum algorithm efficient. In particular, although the majorization principle [19] suggests a local navigation, it does not specify the properties of the global performance landscape of the algorithm that could make it efficient.

By contrast, our approach proposes to explain quantum entanglement in terms of the nonlocal correlations determined by the self-organization processes of prime integer relations. Moreover, according to the description the wave function of a system encodes information about the self-organization processes of prime integer relations the system is defined by [9]. Furthermore, the computational experiments raise the possibility that following the one and the same direction of the processes, the global performance landscape of an algorithm can be made remarkably concave for the algorithm to become efficient.

To have a connection with the quantum case the average distance produced by the algorithm $\mathcal{A}$ solving a TSP problem can be written as a function of the control parameter ϑ

$$\begin{aligned}\bar{D}(\vartheta) = \frac{1}{n}(&\gamma_{1,\ldots,N-1}(\vartheta)d([1,\ldots,N-1>) + \ldots \\ &+\gamma_{N-1,\ldots,1}(\vartheta)d([N-1,\ldots,1>)),\end{aligned}$$

where $\gamma_{i_1,\ldots,i_{N-1}}(\vartheta)$ is the number of agents using the route $[i_1,\ldots,i_{N-1}>$,

$$d([i_1,\ldots,i_{N-1}>)$$

is the distance of the route and the N cities of the problem are labeled by $0,1,\ldots,N-1$ with 0 for the initial city. The interpretation of the coefficient

$$\frac{\gamma_{i_1,\ldots,i_{N-1}}(\vartheta)}{n}$$

as the probability of the route $[i_1,\ldots,i_{N-1}>$ may reduce the minimization of the average distance in the algorithm $\mathcal{A}$ to the maximization of the probability of the shortest routes to be measured in a quantum algorithm.

Moreover, common features of the algorithm $\mathcal{A}$ and Shor's algorithm for integer factorization [20] have been also identified [15].

5 Conclusion

In the paper we have considered the hierarchical network of prime integer relations as a system of information systems and suggested the integration principle as the master equation of the dynamics of an information system in the hierarchical network.

Remarkably, once the integration principle of an information system is realized, the geometrical pattern of the system could take the shape of the geometrical pattern of the problem, while the structures of the information system and the problem would become identical.

We hope that the integration principle could open a new way to solve complex problems efficiently [21, 22].

References

1. Korotkikh, V.: Integer Code Series with Some Applications in Dynamical Systems and Complexity. Computing Centre of the Russian Academy of Sciences, Moscow (1993)
2. Korotkikh, V.: A symbolic description of the processes of complex systems. J. Comput. Syst. Sci. Int. **33**, 16–26 (1995) translation from Izv. Ross. Akad. Nauk, Tekh. Kibern, **1**, 20–31 (1994)
3. Korotkikh, V.: A Mathematical Structure for Emergent Computation. Kluwer Academic Publishers, Dordrecht (1999)
4. Korotkikh, V., Korotkikh, G.: Description of complex systems in terms of self-organization processes of prime integer relations. In: Novak, M.M. (ed.) Complexus Mundi: Emergent Patterns in Nature, pp. 63–72. World Scientific, New Jersey (2006). Available via arXiv:nlin/0509008
5. Korotkikh, V.: Towards an irreducible theory of complex systems. In: Pardalos, P., Grundel, D., Murphey, R., Prokopyev, O. (eds.) Cooperative Networks: Control and Optimization, pp. 147–170. Edward Elgar Publishing, Cheltenham (2008)
6. Korotkikh, V., Korotkikh, G.: On irreducible description of complex systems. Complexity **14**(5) 40–46 (2009)
7. Korotkikh, V., Korotkikh, G.: On an irreducible theory of complex systems. In: Minai, A., Braha, D., Bar-Yam, Y. (eds.) Unifying Themes in Complex Systems, pp. 19–26. Springer: Complexity, New England Complex Systems Institute book series, Berlin (2009)
8. Korotkikh, V.: Arithmetic for the unification of quantum mechanics and general relativity. J. Phys. Conf. **174**, 012055 (2009)
9. Korotkikh, V.: Integers as a key to understanding quantum mechanics. In: Khrennikov, A. (ed.) Quantum Theory: Reconsideration of Foundations - 5, pp. 321–328. AIP Conference Proceedings, vol. 1232, New York (2010)
10. Korotkikh, V.: On possible implications of self-organization processes through transformation of laws of arithmetic into laws of space and time. arXiv:1009.5342v1
11. Einstein, A.: Relativity: The Special and the General Theory - A Popular Exposition. Routledge, London (1960)
12. Korotkikh, G., Korotkikh, V.: On the role of nonlocal correlations in optimization. In: Pardalos, P., Korotkikh, V. (eds.) Optimization and Industry: New Frontiers, pp. 181–220. Kluwer Academic Publishers, Dordrecht (2003)
13. Korotkikh, V., Korotkikh, G., Bond, D.: On optimality condition of complex systems: computational evidence. arXiv:cs.CC/0504092.

14. Korotkikh, V., Korotkikh, G.: On a new type of information processing for efficient management of complex systems. InterJournal of Complex Systems, 2055 (2008) Available via arXiv/0710.3961
15. Korotkikh, V., Korotkikh, G.: On principles in engineering of distributed computing systems. Soft Computing. **12**(2), 201–206 (2008)
16. Korotkikh, V., Korotkikh, G.: Complexity of a system as a key to its optimization. In: Pardalos, P., Grundel, D., Murphey, R., Prokopyev, O. (eds.) Cooperative Networks: Control and Optimization, pp. 171–186. Edward Elgar Publishing, Cheltenham (2008)
17. Korotkikh, V.: Multicriteria analysis in problem solving and structural complexity. In: Pardalos, P., Siskos, Y., Zopounidis, C. (eds.) Advances in Multicriteria Analysis, pp. 81–90. Kluwer Academic Publishers, Dordrecht (1995)
18. Gisin, N.: Can relativity be considered complete? From Newtonian nonlocality to quantum nonlocality and beyond. arXiv:quant-ph/0512168
19. Orus, R., Latorre, J., Martin-Delgado, M. A.: Systematic analysis of majorization in quantum algorithms. arXiv:quant-ph/0212094
20. Maity, K., Lakshminarayan, A.: Quantum chaos in the spectrum of operators used in Shor's algorithm. arXiv:quant-ph/0604111
21. Korotkikh, V., Korotkikh, G.: On principles of developing and functioning of the cyber infrastructure for the Australian coal industry. Coal Supply Chain Cyber Infrastructure Workshop, Babcock & Brown Infrastructure, Level 25, Waterfront Place, Brisbane, August 15 (2006)
22. Korotkikh, G., Korotkikh, V.: From space and time to a deeper reality as a possible way to solve global problems. In: Sayama, H., Minai, A.A., Braha, D., Bar-Yam, Y. (eds.) Unifying Themes in Complex Systems, vol. VIII, pp. 1565–1574. New England Complex Systems Institute Series on Complexity, NECSI Knowledge Press (2011) Available via arXiv:1105.0505v1

On the Optimization of Information Workflow

Michael J. Hirsch, Héctor Ortiz-Peña, Rakesh Nagi, Moises Sudit, and Adam Stotz

Abstract Workflow management systems allow for visibility, control, and automation of some of the business processes. Recently, nonbusiness domains have taken an interest in the management of workflows and the optimal assignment and scheduling of workflow tasks to users across a network. This research aims at developing a rigorous mathematical programming formulation of the workflow optimization problem. The resulting formulation is nonlinear, but a linearized version is produced. In addition, two heuristics (a decoupled heuristic and a greedy randomized adaptive search procedure (GRASP) heuristic) are developed to find solutions quicker than the original formulation. Computational experiments are presented showing that the GRASP approach performs no worse than the other two approaches, finding solutions in a fraction of the time.

Keywords Workflow optimization • Decomposition heuristic • GRASP • Nonlinear mathematical program

M.J. Hirsch (✉)
Raytheon Company, Intelligence and Information Systems, 300 Sentinel Drive, Annapolis Junction, MD 20701, USA
e-mail: Michael.Hirsch@Raytheon.com

H. Ortiz-Peña • M. Sudit • A. Stotz
CUBRC, 4455 Genesee Street, Buffalo, NY 14225, USA
e-mail: Hector.Ortiz-Pena@cubrc.org; Sudit@cubrc.org; Stotz@cubrc.org

R. Nagi
Department of Industrial and Systems Engineering, University at Buffalo, 438 Bell Hall, Buffalo, NY 14260, USA
e-mail: Nagi@buffalo.edu

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_3,

1 Introduction

In general, a workflow management system (WfMS) allows for control and assessment of the tasks (or activities) associated with a business process, defined in a workflow. A workflow is a model of a process, consisting of a set of tasks, users, roles, and a control flow that captures the interdependencies among tasks. The control flow can be defined explicitly by indicating precedence relationships among the tasks, or indirectly by the information requirements (e.g., documents, messages) in order to perform the tasks. WfMS has emerged as an important technology for automating business processes, drawing increasing attention from researchers. Ludascher et al. [9] provide a thorough introduction to workflows and present a few scientific workflow examples. Georgakopoulos et al. [7] discussed three different types of workflows: ad hoc, administrative, and production. Ad hoc workflows perform standard office processes, where there is no set pattern for information flow across the workflow. Administrative workflows involve repetitive and predictable business processes, such as loan applications or insurance claims. Production workflows, on the other hand, typically encompass a complex information process involving access to multiple information systems. The ordering and coordination of tasks in such workflows can be automated. However, automation of production workflows is complicated due to: (a) information process complexity, and (b) accesses to multiple information systems to perform work and retrieve data for making decisions (to contrast, administrative workflows rely on humans for most of the decisions and work performed). WfMSs that support production workflow must provide facilities to define task dependencies and control task execution with little or no human interaction. Production WfMSs are often mission critical in nature and must deal with the integration and interoperability of heterogeneous, autonomous, and/or distributed information systems.

There are many different items to consider with WfMS. One key aspect is the optimal assignment and scheduling of the tasks in a workflow. Joshi [8] discussed the problem of workflow scheduling aiming to achieve cost reduction through an optimal assignment and scheduling of workflows. Each workflow was characterized by a unique due date and tardiness penalty. The problem is formulated as a mixed integer linear program (MILP). The model assumed that the dependencies and precedence relationships among the workflows are deterministic and unique. Tasks are not preemptive and the processing times and due dates are also deterministic and known. Users can assume several roles but can perform only one task at a time. The total cost component which the model tries to minimize consists of two elements: *processing cost* and *tardiness penalty cost*. Processing cost refers to the price charged by the user to perform the assigned tasks; tardiness penalty cost refers to the product of a unit tardiness penalty for the workflow and the time period by which it is late (with respect to its assigned due date). A branch and price approach was proposed to solve the problem. Moreover, an acyclic graph heuristic was developed to solve the sub-problems of this approach. The proposed algorithm was used to solve static and reactive situations. Reactive scheduling (or rescheduling) is

the process of revising a given schedule due to unexpected events. Events considered by the author included: change in priority of a workflow, change in processing time of tasks, and the addition of new workflows. The results of a computational study indicate the benefits of using reactive strategies when the magnitude and frequency of changes increase. Nukala [10] described the software implementation details (e.g., architecture, data files manipulation, etc.) while developing the schedule deployer and POOL (Process Oriented OpenWFE Lisp) updater (SDPU) application which uses the algorithm described in Joshi [8] as the workflow scheduler.

Xiao et al. [15] proposed an optimization method of workflow pre-scheduling based on a nested genetic algorithm (NGA). By pre-scheduling, the authors refer to the scheduling of all tasks when the workflow is initialized (as opposed to, e.g., reactive workflow scheduling in which tasks might be scheduled even when the workflow is active and some tasks have already been completed). The problem can then be described as finding the optimal precedence and resource allocation such that the finish time of the last task is minimized. NGA uses nested layers to optimize several variables. For this approach, two variables were considered: an inner layer referring to the allocation of resources and an outer layer referring to the execution sequence of tasks. The solutions found by the NGA algorithm were better than the solutions found by a dynamic method consisting of a greedy heuristic that assigned the resource able to complete the task fastest to execute the task. Dewan et al. [2] presented a mathematical model to optimally consolidate tasks to reduce the overall cycle time in a business information process. Consolidation of tasks may reduce or eliminate cost losses and delays due to the required communication and information hand-off between tasks. On the other hand, consolidation represents loss of specialization, which may result in larger process time. Using this formulation, the authors analytically and numerically present the impact of delay costs, hand-off, and loss of specialization on the benefits of tasks consolidation.

In Zhang et al. [17], the authors considered quality-of service (QoS) optimization, by minimizing the number of machines, subject to customer response time and throughput requirements. They propose an efficient algorithm that decomposes the composite-service level response time requirements into atomic-service level response time requirements that are proportional to the CPU consumption of the atomic services. Binary search was incorporated in their algorithm to identify the maximum throughput that can be supported by a set of machines. A greedy algorithm was used for the deployment of services across machines. Zhang et al. [18] presented research on grid workflow and dynamic scheduling. The "grid" refers to a new computing infrastructure consisting of large-scale resource sharing and distributed system integration. Grid workflow is similar to traditional workflow but most of grid applications are, however, high performance computing and data intensive requiring efficient, adaptive use of available grid resources. The scheduling of a workflow engine has two types: static scheduling and dynamic scheduling. The static scheduling allocates needed resources according to the workflow process description. The dynamic scheduling also allocates resources according to the process description but takes into account the conditions of grid resources. Although

the authors indicated that the resources can be scheduled according to QoS and performance, the algorithm only considers the latter. The grid workflow modeling on this research is based on petri nets.

Xianwen et al. [14] presented a dependent task static scheduling problem considering the dynamics and heterogeneity of the grid resources. A reduced task-resource assignment graph (RT-RAG)-based scheduling model and an immune generic algorithm (IGA) scheduling algorithm were proposed. The RT-RAG is expressed as a 4-tuple $< V, E, WV, WE >$ in which V represents the set of nodes, E represents the set of precedence constraints, WV represents the node weight, and WE represents the edges weight. The nodes express a mapping from a task to a resource, the node weights express communication data, and the edge weights include the computation cost and the bandwidth constraints. The RT-RAG is derived from the task graph (including all tasks) by applying a "reduction rule" which removes all tasks that are finished. The proposed IGA performed better than the adaptive heterogeneous earliest finish time (HEFT)-based Rescheduling (AHEFT) algorithm [16] and the dynamic scheduling algorithm Min-Max [11] in the experiments conducted by the authors. It was indicated in the paper that the initial parameters of IGA were critical to the performance of the algorithm. Tao et al. [13] proposed and evaluated the performance of the rotary hybrid discrete particle swarm optimization (RHDPSO) algorithm for the multi-QoS constrained grid workflow scheduling problem. This grid system selects services from candidate services according to the parameters set by a user to complete the grid scheduling. QoS was defined as a six-tuple (Time, Cost, Reliability, Availability, Reputation, Security) to characterize the service quality. Time measures the speed of a service response. This is expressed as the sum of the service execution time and service communication time. Cost describes the total cost of service execution. Reliability indicates the probability of the service being executed successfully. Availability measures the ability of the service to finish in the prescriptive time. The value of availability is computed using the ratio of service execution time and the prescriptive time. Reputation is a measure of service "trustworthiness" and it is expressed as the average ranking given by users. Security is a measure indicating the possibility of the service being attacked or damaged. The higher this value, the lower this possibility. The advantage of the RHDPSO algorithm over a discrete particle swarm optimization algorithm is its speed of convergence and the ability to obtain faster and feasible schedules.

In our research, we are concerned with information workflows. Information is generated by some tasks (i.e., produced as output) and consumed by other tasks (i.e., needed as input). There are multiple information workflows, with multiple tasks, that need to be assigned to users. Users can take on certain roles, and tasks can only be performed by users with certain roles. The tasks themselves can have precedence relationships. Overall, the goal is to minimize the time at which the last task gets completed. We formulated the assignment and scheduling of tasks to users, and the information flow amongst the users as a mixed-integer nonlinear program (MINLP). The flow of information considered the required information by certain

tasks (as input), the information produced by tasks (as output), and precedence relationships between tasks. In addition to linearizing the MINLP, two heuristics were developed; a decomposition heuristic and the construction phase of a greedy randomized adaptive search procedure (GRASP). The rest of this paper is organized as follows: Sect. 2 describes the MINLP formulation for the information workflow problem, as well as the linearization. In Sect. 3, a detailed description of the two heuristic approaches considered to solve the MILNP is provided. A computational study and analysis is presented in Sect. 4. Finally, conclusions and future research are discussed in Sect. 5.

2 Mathematical Formulations

2.1 Problem Definition

The overall problem addressed here is to assign tasks occurring on multiple information workflows to users, and flow information amongst the users, from tasks that produce information as output to tasks that require the information as input. Each task can only be assigned to a user if the task roles and the user roles overlap. Users have processing times to accomplish tasks. There are precedence relationships between the tasks (e.g., Task k must be completed before Task m can start). In addition, tasks might require certain information as input before they can begin, and tasks might generate information as output when they are completed. If a task needs a certain piece of information as input (e.g., α), there needs to be an assignment of the transference of α from a user assigned a task producing α as output to the user assigned the task requiring α as input. We are assuming in this research that the transference of information from one user to another is instantaneous, i.e., that there is infinite bandwidth. In the sequel we will consider the case of finite uplink and downlink bandwidth. We note that we use the term "user" rather loosely. A user could be an human performing a task, as well as an automated system performing a task. We also make the assumption that two tasks assigned to one user cannot be accomplished simultaneously, i.e., once a user starts one task, that task needs to be completed before the user can move on to the next task it is assigned.

Figure 1 presents an example scenario. In this scenario, there are two workflows. The goal is to assign the tasks on the workflows to users, schedule the tasks assigned to each user, and determine the appropriate information transfers that need to take place among the users, and when those information transformations need to be performed. In this figure, the tasks are color-coded by the user assignments, and the information flows detailed. In this example scenario, *User 1* is first assigned to perform *Task 1* (on workflow 1). When *Task 1* is completed, and *User 1* receives information μ from *User 3*, then *User 1* can begin its next task, *Task 3* (on workflow 2). Once complete with *Task 3*, *User 1* can begin *Task 8* (on workflow 2).

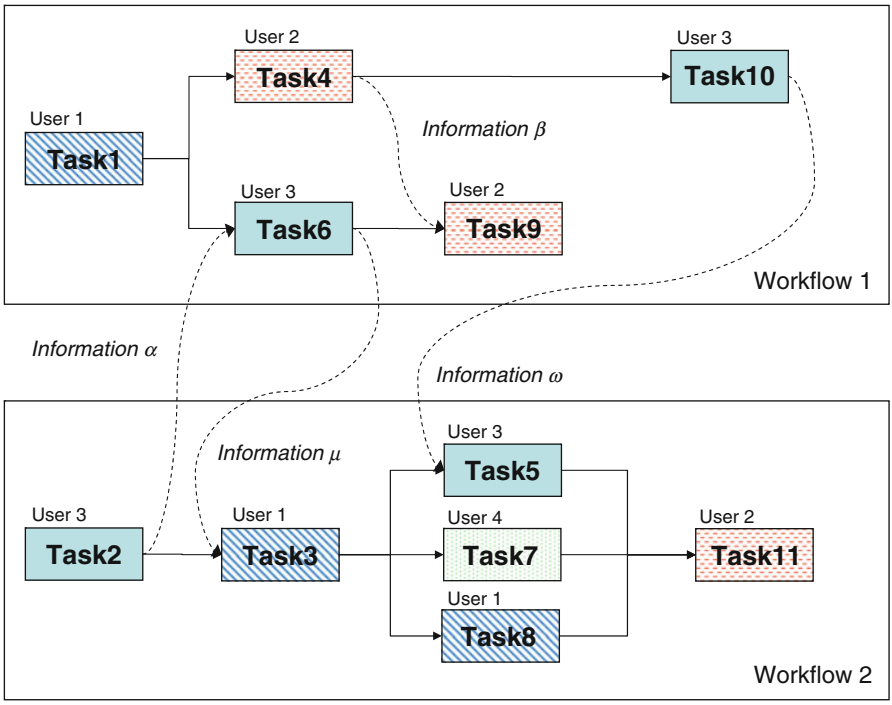

Fig. 1 Multiple information workflows, with users assigned to tasks, and information flow defined across users

2.2 Parameters

This section introduces the parameters incorporated into the information workflow optimization formulation. For the mathematical formulation to follow, the parameters, decision variables, and constraints are defined for all $i \in \{1, \ldots, \mathcal{T}\}$, $j \in \{1, \ldots, \mathcal{P}\}$, $q \in \{1, \ldots, \mathcal{Q}\}$, and $n \in \{1, \ldots, \mathcal{N}\}$. (*N.B.*: It is possible for $\mathcal{N}$ to be 0; the case when there is no information artifacts produced as possible outputs of some tasks and/or inputs of other tasks. In that case, all constraints and decision variables that use n as an index drop out of the nonlinear and linearized formulations.)

$\mathcal{T} \in \mathbb{Z}_+$ defines the number of activities (indexed by i).
$\mathcal{P} \in \mathbb{Z}_+$ defines the number of users (indexed by j).
$\rho_{ij} \in \mathbb{R}_+ \cup \{0\}$ defines the processing time of activity i by user j.
$\mathcal{Q} \in \mathbb{Z}_+$ defines the number of possible roles (indexed by q).
$\mathcal{N} \in \mathbb{Z}_+ \cup \{0\}$ defines the number of possible inputs / outputs of all activities on all workflows—called information artifacts (indexed by n).

R_j is binary vector of length $\mathcal{Q}$, where $R_{jq} = \begin{cases} 1 & \text{if user } j \text{ can perform role } q \\ 0 & \text{o.w.} \end{cases}$.

$\bar{R}_i$ is binary vector of length $\mathcal{Q}$, where $\bar{R}_{iq} = \begin{cases} 1 & \text{if activity } i \text{ can be performed by role } q \\ 0 & \text{o.w.} \end{cases}$.

I_i is binary vector of length $\mathcal{N}$, where $I_{in} = \begin{cases} 1 & \text{if activity } i \text{ requires information artifact } n \\ 0 & \text{o.w.} \end{cases}$.

O_i is binary vector of length $\mathcal{N}$, where $O_{in} = \begin{cases} 1 & \text{if activity } i \text{ produces information artifact } n \\ 0 & \text{o.w.} \end{cases}$.

V_i is a binary vector of length $\mathcal{T}$, where $V_{i\hat{\imath}} = \begin{cases} 1 & \text{if activity } \hat{\imath} \text{ must finish before activity } i \text{ can start} \\ 0 & \text{o.w.} \end{cases}$.

$\mathcal{H}$ is a large enough constant number.

2.3 Decision Variables

This section defines the main decision variables for the mathematical formulation.

$y_{ij} = \begin{cases} 1 & \text{if activity } i \text{ is assigned to user } j \\ 0 & \text{o.w.} \end{cases}$.

$Y_j =$ Number of activities assigned to user j.

$z_{ij\ell} = \begin{cases} 1 & \text{if } \ell\text{-th activity of user } j \text{ is activity } i \\ 0 & \text{o.w.} \end{cases}$.

$\hat{z}_{j\ell} =$ The number of activities assigned to be the ℓ-th activity of user j. N.B.: This is used to enforce the ℓ-th activity of user j not being assigned if the $(\ell-1)$-th activity of user j is not assigned.

$\hat{S}_i =$ Start time of activity i.

$\hat{E}_i =$ End time of activity i.

$S_{j\ell} =$ Start time of ℓ-th activity of user j.

$E_{j\ell} =$ End time of ℓ-th activity of user j.

$\Phi_{\hat{\jmath}jn} = \begin{cases} 1 & \text{if user} \hat{\jmath} \text{ is assigned to send information artifact } n \text{ to user } j \\ 0 & \text{o.w.} \end{cases}$.

2.4 Nonlinear Formulation

The resultant nonlinear formulation is given as (where, in all constraints to follow, $i, \hat{\imath} \in \{1, \ldots, \mathcal{T}\}, i \neq \hat{\imath}$, $j, \hat{\jmath} \in \{1, \ldots, \mathcal{P}\}$, and $\ell, \hat{\ell} \in \{1, \ldots, \mathcal{T}\}$):

$$F = \min \left[\max_i \{\hat{E}_i\} \right] \tag{1}$$

s.t.

$$y_{ij} \leq \sum_{q=1}^{\mathcal{Q}} R_{jq} \bar{R}_{iq} \ \forall i, j, \tag{2}$$

$$\sum_{j=1}^{\mathcal{P}} y_{ij} = 1 \ \forall i, \tag{3}$$

$$\sum_{\ell=1}^{\mathcal{T}} z_{ij\ell} = y_{ij} \ \forall i, j, \tag{4}$$

$$\hat{S}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} z_{ij\ell} S_{j\ell} \ \forall i, \tag{5}$$

$$\hat{E}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} z_{ij\ell} E_{j\ell} \ \forall i, \tag{6}$$

$$E_{j\ell} = S_{j\ell} + \sum_{i=1}^{\mathcal{T}} \rho_{ij} z_{ij\ell} \ \forall \ell, j, \tag{7}$$

$$E_{j0} = 0 \ \forall j, \tag{8}$$

$$S_{j\ell} \geq E_{j,\ell-1} \ \forall \ell, j, \tag{9}$$

$$\hat{S}_i \geq V_{i\hat{\imath}} \hat{E}_{\hat{\imath}} \ \forall i, \hat{\imath}, \tag{10}$$

$$S_{j\ell} \geq I_{in} z_{ij\ell} \Phi_{\hat{\jmath} jn} z_{\hat{\imath}\hat{\jmath}\hat{\ell}} O_{\hat{\imath}n} E_{\hat{\jmath}\hat{\ell}} \ \forall n, \ell, \hat{\ell}, j, \hat{\jmath}, i, \hat{\imath}, \tag{11}$$

$$\sum_{\hat{\jmath}=1}^{\mathcal{P}} \Phi_{\hat{\jmath} jn} \geq I_{in} y_{ij} \ \forall n, i, j, \tag{12}$$

$$\Phi_{\hat{\jmath} jn} \leq \sum_{\hat{\imath}=1}^{\mathcal{T}} O_{\hat{\imath}n} y_{\hat{\imath}\hat{\jmath}} \ \forall n, j, \hat{\jmath}, \tag{13}$$

$$\Phi_{\hat{\jmath} jn} \leq \sum_{i=1}^{\mathcal{T}} I_{in} y_{ij} \ \forall n, j, \hat{\jmath}, \tag{14}$$

$$\sum_{\hat{j}=1}^{\mathcal{P}} \Phi_{\hat{j}jn} \leq 1 \ \forall n, j, \tag{15}$$

$$\hat{z}_{j\ell} = \sum_{i=1}^{\mathcal{T}} z_{ij\ell} \ \forall j, \ell, \tag{16}$$

$$\hat{z}_{j\ell} \geq \hat{z}_{j,\ell+1} \ \forall \ell \in \{1, \ldots, \mathcal{T}-1\}, j, \tag{17}$$

$$\sum_{i=1}^{T} z_{ij\ell} \leq 1 \ \forall j, \ell, \tag{18}$$

$$\hat{S}_i, \hat{E}_i, S_{j\ell}, E_{j\ell} \in [0, \mathcal{H}] \ \forall \ell, j, i, \tag{19}$$

$$\hat{z}_{j\ell}, y_{ij}, z_{ij\ell}, \Phi_{\hat{j}jn} \in \{0, 1\} \ \forall n, \ell, \hat{j}, j, i, \tag{20}$$

Interpretation of Formulation

The objective function (1) minimizes the maximum time at which an activity is completed.
Constraint (2) only permits user j to perform activity i if there exists some commonality between the roles user j can assume and the roles necessary to fill activity i.
Constraint (3) enforces that each activity must be assigned exactly one user.
Constraint (4) specifies that one of the ℓ activities of user j must be activity i, if and only if activity i is assigned to user j.
Constraint (5) relates the start time of activity i to the start time of the ℓ-th activity of user j.
Constraint (6) relates the end time of activity i to the end time of the ℓ-th activity of user j.
Constraint (7) relates the end time of the ℓ-th activity of user j to the start time of the ℓ-th activity and the processing time of the activity.
Constraint (8) provides initial values for variables needed to make Constraint (9) consistent.
Constraint (9) relates the starting time of the ℓ-th of user j to the ending time of the $(\ell-1)$-th activity of user j.
Constraint (10) enforces that the starting time of activity i must occur after activity $\hat{\imath}$ is completed, if there exists a precedence relationship between activities i and $\hat{\imath}$.
Constraint (11) enforces that the starting time of the ℓ-th activity of user j must occur after the $\hat{\ell}$-th activity of user $\hat{\jmath}$ is completed, if activity i is the ℓ-th activity of user j, activity $\hat{\imath}$ is the $\hat{\ell}$-th activity of user $\hat{\jmath}$, activity i needs as input information artifact n, activity $\hat{\imath}$ produces information artifact n as output, and user $\hat{\jmath}$ is assigned to send information artifact n to user j.
Constraint (12) will force at least one user to send information artifact n to user j if user j needs it as input for an assigned task. *N.B.:* This constraint, by itself, does

not verify that the user assigned to send information artifact n to user j does in fact have a task that produces information artifact n as output.
Constraint (13) will force $\Phi_{\hat{\jmath}jn}$ to be 0 if user $\hat{\jmath}$ is not assigned a task that produces information artifact n as output.
Constraint (14) will force $\Phi_{\hat{\jmath}jn}$ to be 0 if user j is not assigned a task that needs information artifact n as input.
Constraint (15) allows at most one user to send information artifact n to user j.
Constraints (16) and (17) allow the $\ell + 1$-th activity of user j to be assigned an activity only if the ℓ-th activity of user j is also assigned.
Constraints (18) permit at most one activity to be assigned as the ℓ-th activity of user j.
Constraints (19) and (20) prescribe domain restrictions on all of the decision variables.

N.B.: From the parameters section, $\mathcal{H}$ is defined to be a "large-enough" constant. Determining large enough is in most cases more of an art than a science. However, there are some simple methods to determine analytically appropriate values for these types of constants. We know that the worst-case scenario would be to have the activities done serially, one after the other, with the user assigned to each activity being the user that would take the most amount of time to perform that activity. Hence, an appropriate upper bound for $\mathcal{H}$ would be

$$\mathcal{H} = \sum_{i=1}^{\mathcal{T}} \max_{j=1,\ldots,\mathcal{P}} \left[\rho_{ij}\right]. \tag{21}$$

2.5 *Linearized Formulation*

We can linearize the mixed-integer nonlinear programming formulation (1)–(20) as follows. To begin with, we can replace the objective function (1) with minimizing ξ and adding the additional constraints

$$\xi \geq \hat{E}_i \ \forall i, \tag{22}$$

$$\xi \in [0, \mathcal{H}]. \tag{23}$$

We replace (5) with

$$\hat{S}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} \mu_{ij\ell} \ \forall i \tag{24}$$

adding the additional constraints

$$\mu_{ij\ell} \leq S_{j\ell} \ \forall i, j, \ell, \tag{25}$$

$$\mu_{ij\ell} \geq S_{j\ell} - \left(1 - z_{ij\ell}\right) \mathcal{H} \ \forall i, j, \ell, \tag{26}$$

$$\mu_{ij\ell} \leq \mathcal{H} z_{ij\ell} \ \forall i, j, \ell, \tag{27}$$

$$\mu_{ij\ell} \geq 0 \ \forall i, j, \ell. \tag{28}$$

We replace (6) with

$$\hat{E}_i = \sum_{j=1}^{\mathcal{P}} \sum_{\ell=1}^{\mathcal{T}} v_{ij\ell} \ \forall i \tag{29}$$

adding the additional constraints

$$v_{ij\ell} \leq E_{j\ell} \ \forall i, j, \ell, \tag{30}$$

$$v_{ij\ell} \geq E_{j\ell} - \left(1 - z_{ij\ell}\right) \mathcal{H} \ \forall i, j, \ell, \tag{31}$$

$$v_{ij\ell} \leq \mathcal{H} z_{ij\ell} \ \forall i, j, \ell, \tag{32}$$

$$v_{ij\ell} \geq 0 \ \forall i, j, \ell. \tag{33}$$

Finally, we replace (11) with

$$S_{j\ell} \geq I_{in} O_{\hat{\imath}n} \delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \ \forall n, \ell, \hat{\ell}, j, \hat{\jmath}, i, \hat{\imath} \tag{34}$$

and add the constraints

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq z_{ij\ell} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, \tag{35}$$

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq z_{\hat{\imath}\hat{\jmath}\hat{\ell}} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, \tag{36}$$

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \geq z_{ij\ell} + z_{\hat{\imath}\hat{\jmath}\hat{\ell}} - 1 \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, \tag{37}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq E_{\hat{\jmath}\hat{\ell}} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, \tag{38}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \geq E_{\hat{\jmath}\hat{\ell}} - \left(1 - \alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}}\right) \mathcal{H} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, \tag{39}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \leq \mathcal{H} \alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, \tag{40}$$

$$\gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \geq 0 \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, \tag{41}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \leq \gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n, \tag{42}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \geq \gamma_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} - \left(1 - \Phi_{\hat{\jmath}jn}\right) \mathcal{H} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n, \tag{43}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \leq \Phi_{\hat{\jmath}jn} \mathcal{H} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n, \tag{44}$$

$$\delta_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}n} \geq 0 \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}, n, \tag{45}$$

$$\alpha_{i\hat{\imath}j\hat{\jmath}\ell\hat{\ell}} \in \{0, 1\} \ \forall i, \hat{\imath}, j, \hat{\jmath}, \ell, \hat{\ell}. \tag{46}$$

Table 1 Comparison on the number of decision variables and constraints for the nonlinear formulation and the linearized formulation, in terms of the parameters $\mathcal{T}, \mathcal{P}, \mathcal{N}$

Problem	Decision variables	Constraints
Nonlinear	$O\left(\mathcal{T}^2\mathcal{P} + \mathcal{P}^2\mathcal{N}\right)$	$O\left(\mathcal{T}^4\mathcal{P}^2\mathcal{N} + \mathcal{T}^2\mathcal{P}\right)$
Linearized	$O\left(\mathcal{T}^4\mathcal{P}^2\mathcal{N} + \mathcal{T}^2\mathcal{P}\right)$	$O\left(\mathcal{T}^4\mathcal{P}^2\left(\mathcal{N}+1\right)\right)$

Thus, the problem becomes one of minimizing ξ subject to the constraints (2)–(4), (7)–(10), (12)–(20), and (22)–(46). The resulting formulation is a MILP and can be solved using a number of commercial software packages. Table 1 compares the number of decision variables and constraints for the nonlinear formulation of Sect. 2.4 and the linearized formulation.

2.6 Complexity of Formulation

Since our problem formulation is an extension of the extensively studied job scheduling problem, which has been demonstrated to be `NP-complete` [6], our problem is `NP-complete` as well. There is no known efficient algorithm to find a solution that will guarantee optimality to these kinds of problems in polynomial-time. The computational time required to solve `NP-complete` problems increases dramatically as the size of the problem increases. Efficient heuristics that find good-quality solutions are required.

3 Heuristic Approaches

As the problem formulation can be written as a mixed-integer linear program, there are numerous commercial software packages that can be utilized to attempt to solve problem instances. However, as discussed in Sect. 2.6, the problem formulation is `NP-complete`, so the development of heuristics is necessitated as the problem instances grow in size. We have developed two heuristics to find good-quality solutions to the problem.

3.1 Decomposition Heuristic

This section will describe a simple decomposition heuristic to solve the mathematical formulation. The main idea is to decouple the problem of assigning tasks to users from the problem of scheduling the tasks of each user and determining the routing of information among the users. This translates into first

Fig. 2 High-level pseudo-code for GRASP

```
procedure GRASP(Problem Instance)
1     InputInstance();
2     while Stopping criteria not met do
3           ConstructGreedyRandomizedSolution(Solution);
4           LocalSearch(Solution);
5           if Solution better than BestSolution then
6                 UpdateBestSolution(Solution,BestSolution);
7           end if
8     end while
9     return(BestSolution);
end GRASP;
```

minimizing (47), subject to the constraints (2), (3), and then, using the assignment variables y_{ij} as parameters, minimizing (1), subject to constraints (4)–(19).

$$F_1 = \min \left[\max_j \left(\sum_{i=1}^{\mathcal{T}} \rho_{ij}\, y_{ij} \right) \right]. \tag{47}$$

Function (47) load balances the tasks assigned to users and can be linearized by replacing (47) with minimizing $\bar{\xi}$ and adding the additional constraints

$$\bar{\xi} \geq \sum_{i=1}^{\mathcal{T}} \rho_{ij}\, y_{ij} \;\; \forall\, j, \tag{48}$$

$$\bar{\xi} \in [0, \mathcal{H}]. \tag{49}$$

As both formulations are mixed-integer linear programs, commercial software packages can be used to find solutions to problem instances.

3.2 GRASP Heuristic

The first use of the GRASP was for the solution of computationally difficult set-covering problems [3]. Since then, the GRASP approach has been successfully applied to an overwhelming number of combinatorial optimization problems [4, 5, 12]. Feo and Resende [3, 4] describe the metaheuristic GRASP as a multi-start local search procedure, where each GRASP iteration consists of two phases, a construction phase and a local search phase. In the construction phase, interactions between greediness and randomization generate a diverse set of good-quality solutions. The local search phase improves upon the solutions found in construction. The best solution found over all of the multi-start iterations is retained as the final solution. Figure 2 provides high-level pseudo-code of the main GRASP algorithm.

In our approach, we implemented only the construction portion of the GRASP heuristic. Starting from an empty schedule for each user, this heuristic iteratively

```
procedure GRASP − Construction( )
1     α ← RandUnif (0, 1);
2     Initialize (y, Y, z, S, E, Vtime, ArtifactProduced, ArtifactTime, TaskStatus);
3     TaskStatus ← Update (TaskStatus);
4     UnAssigned ← {1, 2, . . . , T};
5     while UnAssigned ≠ ∅ do
6           min ← +∞;
7           max ← −∞;
8           for i = 1, . . . , T do
9                 if TaskStatus[i] = 1 then
10                      h_i ← BestAssignment(y, Y, z, S, E, ArtifactTime, Vtime, I);
11                      g_i ← f (h_i);
12                      if min > g_i then min ← g_i;
13                      if max < g_i then max ← g_i;
14                end if
15          end for
16          RCL ← ∅;
17          Threshold ← min + α ∗ (max − min);
18          for i = 1, . . . , T do
19                if TaskStatus[i] = 1 and g_i ≤ Threshold then
20                      RCL ← RCL ∪ {i};
21                end if
22          end for
23          j ← RandomlySelectElement(RCL);
24          TaskStatus[j] = 0;
25          {y, Y, z, S, E} ← Update (y, Y, z, S, E);
26          {ArtifactProduced, ArtifactTime, Vtime} ← Update (ArtifactProduced, ArtifactTime, Vtime);
27          TaskStatus ← Update (TaskStatus);
28          UnAssigned ← UnAssigned \ {j};
29    end while
end GRASP − Construction;
```

Fig. 3 Pseudo-code for GRASP construction procedure

constructs a feasible schedule, combining greediness and randomization at each iteration. The pseudo-code is found in Fig. 3, and we describe it in detail. Parameter initialization gets accomplished in Line 2, based on the specific problem instance data. Line 3 updates the vector `TaskStatus`, based on whether a task is "available" to be assigned. `TaskStatus`$[i]$ is set to 2 if the i-th task is not currently available to be assigned to a user (because of task precedence or input information artifact constraints that are not satisfied), or set to 1 to designate that the i-th task is available for assignment. The while loop in lines 5–29 are executed until all of the tasks are assigned to users. Each time through the while loop, if the status for task i is set to 1, then line 10 determines the best assignment of task i to the users (placing task i at the end of the schedule), taking into account the times that all input information artifacts are available for this task, as well as the ending times of any tasks that have precedence over this task. Line 11 sets g_i to be the time at which all currently assigned tasks (including task i) get completed. After looping through all unfixed coordinates (lines 8–15), in lines 18–22 a restricted candidate list (RCL) is formed containing the assignable tasks i whose g_i values are less than the threshold (defined on line 17). In line 23, a task is chosen at random from the RCL, say task j. At that point, the `TaskStatus` for task j is set to 0, parameters and decision variables are updated. Line 27 updates `TaskStatus`, because as a result of task j

being assigned, tasks that were waiting for an information artifact task j produced as output or had a precedence relationship with task j might now be eligible to be assigned.

4 Computational Experiments

In this section, we compare the solution quality and efficiency of finding solutions to the full formulation, as well as the two heuristics, for a variety of test scenarios.

4.1 Test Environment

The full formulation and the decomposition heuristic were solved using version 12.2.0.0 of `CPLEX` [1]. The GRASP heuristic was implemented in C++, with the Borland C++ compiler. All experiments were conducted on a Hewlett Packard EliteBook 8730w, with 1.59 GHz and 2.96 GB of RAM.

4.2 Experimental Results

To test the three approaches, we simulated multiple workflow processes, varying both the number of users, $\mathcal{P}$, and the number of tasks, $\mathcal{T}$, between 2 and 10. Table 2 lists the values given to parameters which influence how the simulated workflows were created. $\mathcal{N}$ refers to the number of information artifacts potentially present in any given simulation instance. (*N.B.*: just because an information artifact is present, this does not mean it is produced as output nor needed as input from a task on any given workflow). `IO Relation` defines how likely it is for a given task to need as input a certain information artifact, and how likely it is that the task produces the information artifact as output. `Task Precedence` defines how likely a given task is to have a precedence relationship with another task. The `Task Processing Time` defines the interval where the user / task processing times are drawn. For each $\mathcal{T}$, $\mathcal{P}$ combination, we randomly created ten independent experiments, where the parameters in Table 2 were chosen randomly from a uniform distribution defined by the corresponding elements in the second column of the table. We allowed all

Table 2 Parameters used to generate random simulated workflow experiments

Parameter	Value
$\mathcal{N}$	[4, 40]
IO relation	[0, 10]
Task precedence	[0, 10]
Task processing time	[30, 100]

Table 3 (a) Percentage of time decoupled heuristic does no worse than the full formulation; (b) GRASP heuristic does no worse than the full formulation; (c) and GRASP heuristic does no worse than the decoupled heuristic

$\mathcal{T} \setminus \mathcal{P}$		2	3	4	5	6	7	8	9	10
2	(a)	70	60	50	80	80	60	70	60	50
	(b)	100	100	100	100	100	100	100	100	100
	(c)	100	100	100	100	100	100	100	100	100
3	(a)	50	50	70	40	60	20	60	60	40
	(b)	90	80	100	90	90	90	80	100	100
	(c)	100	90	100	90	90	100	80	100	100
4	(a)	90	30	20	40	30	10	20	20	30
	(b)	80	70	80	70	70	90	80	80	90
	(c)	80	90	90	80	90	100	80	90	90
5	(a)	60	50	50	20	20	30	30	60	50
	(b)	40	50	50	90	70	40	90	90	100
	(c)	60	60	70	100	70	60	100	90	100
6	(a)	100	70	50	40	30	70	100	80	100
	(b)	50	70	60	70	60	90	100	90	100
	(c)	50	70	70	80	70	90	80	90	80
7	(a)	100	90	50	70	60	100	100	100	90
	(b)	50	50	60	90	80	100	90	100	100
	(c)	30	50	70	60	80	90	70	100	90
8	(a)	100	100	100	90	100	100	100	100	100
	(b)	60	100	90	100	100	100	100	100	100
	(c)	50	40	50	70	90	100	100	80	80
9	(a)	100	100	100	100	100	100	100	100	100
	(b)	80	100	100	100	100	100	100	100	100
	(c)	50	40	30	70	60	90	100	70	100
10	(a)	100	100	100	100	100	100	100	100	100
	(b)	90	100	100	100	100	100	100	100	100
	(c)	30	10	30	100	100	100	100	100	100

approaches to run for a maximum of 600 seconds. For each experiment, we captured the best solution found by each approach and the time it took for each approach to complete.

Table 3 presents a comparison on solution quality for all three approaches. For each $\mathcal{T}$, $\mathcal{P}$ combination, the row labeled "(a)" lists the percentage of experiments where the decoupled heuristic found a solution no worse than the full formulation, "(b)" lists the percentage of experiments where the GRASP heuristic found a solution no worse than the full formulation, and "(c)" lists the percentage of experiments where the GRASP heuristic found a solution no worse than the decoupled heuristic. It is clear from the table that as $\mathcal{T}$ and $\mathcal{P}$ increase, the GRASP heuristic performs better than either the full formulation and the decoupled heuristic.

Figures 4–12 show the log of the average times each approach needed to find a solution for the experiments. As is clearly evident, the GRASP approach is much

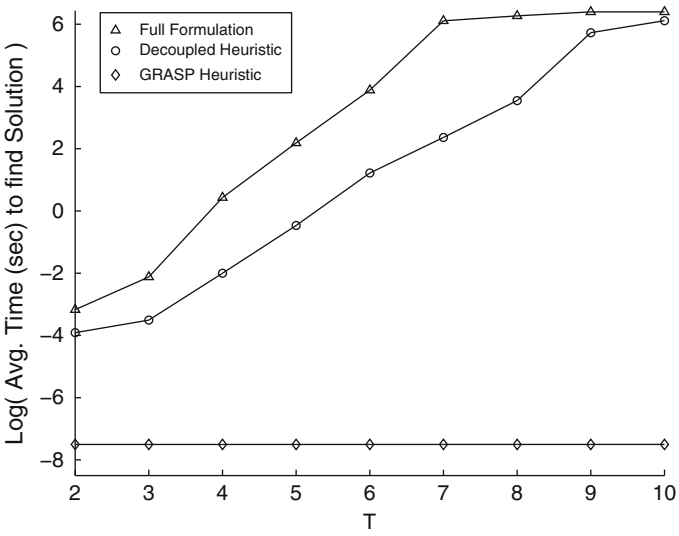

Fig. 4 Log of the average time to find a solution for $\mathcal{P} = 2$

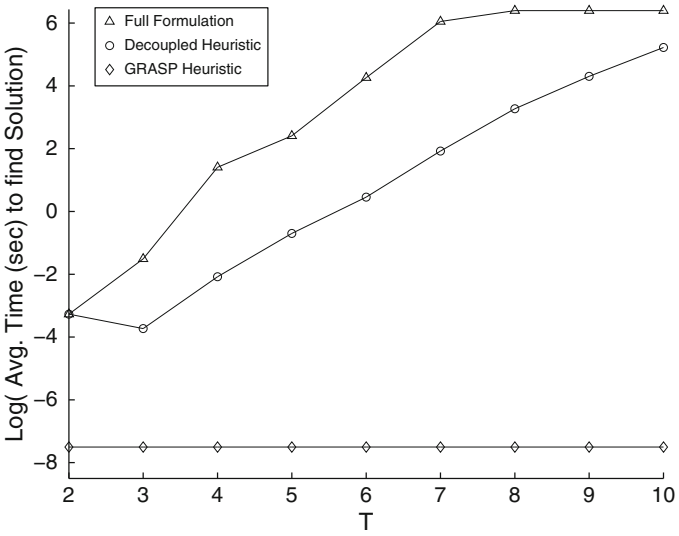

Fig. 5 Log of the average time to find a solution for $\mathcal{P} = 3$

quicker than the full formulation and the decoupled heuristic. In all of these plots, we set -7.5 as the lowest possible value for the log of the solution time. This in effect corresponds with an actual solution time of 0. For almost all of the experiments, the GRASP heuristic registered a solution time of 0 second. The full formulation and the decoupled heuristic, on the other hand, took quite a bit longer, and as $\mathcal{P}$ and $\mathcal{T}$

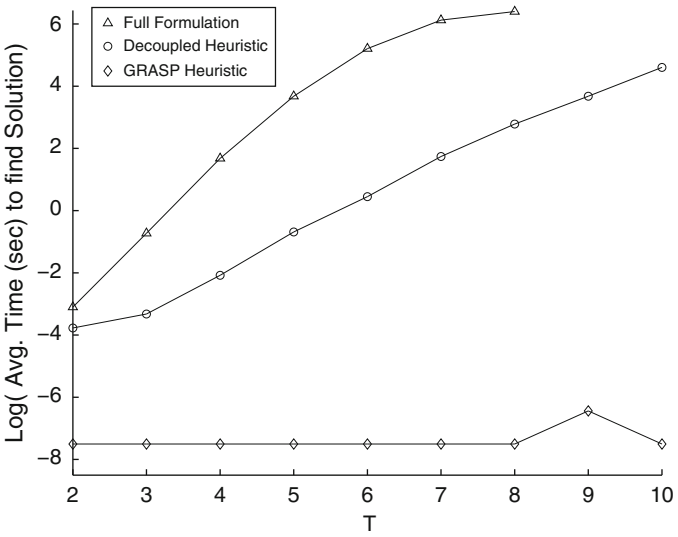

Fig. 6 Log of the average time to find a solution for $\mathcal{P} = 4$

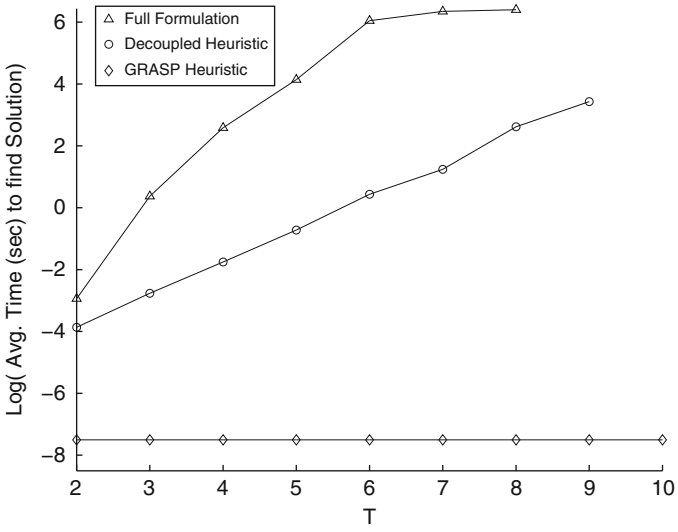

Fig. 7 Log of the average time to find a solution for $\mathcal{P} = 5$

increase, these two approaches are either (a) not able to find a feasible solution in the 600 seconds or (b) CPLEX runs out of memory reading in the problem instance (see Tables 4 and 5). Because the GRASP approach so quickly solves these small problem instances, we also did two quick experiments, looking at first when both $\mathcal{P}$ and $\mathcal{T}$ are set to 100 and then when they are both set to 1000. For the case where

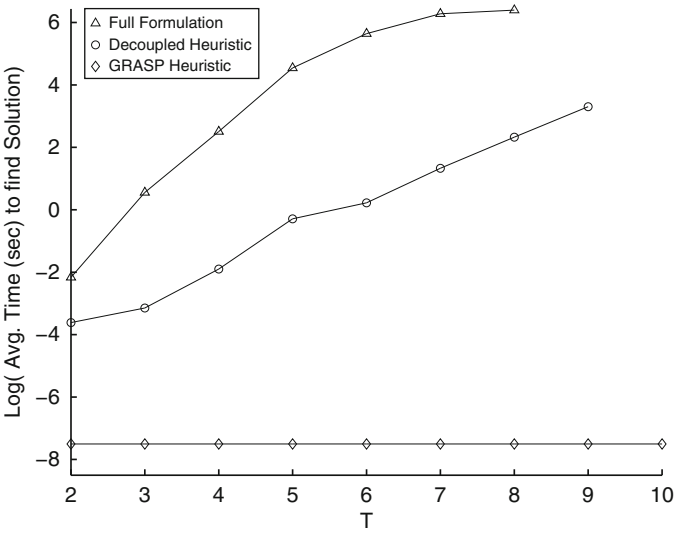

Fig. 8 Log of the average time to find a solution for $\mathcal{P} = 6$

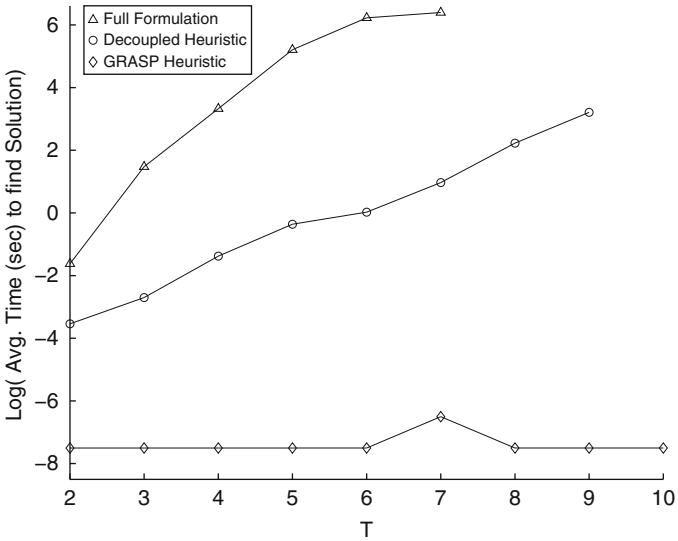

Fig. 9 Log of the average time to find a solution for $\mathcal{P} = 7$

both $\mathcal{P}$ and $\mathcal{T}$ were set to 100, GRASP solved all ten experiments with an average solution time of 0.0486 seconds, while GRASP took an average of 14.9361 seconds to solve all ten experiments for the case when both $\mathcal{P}$ and $\mathcal{T}$ were set to 1000. This gives evidence that the GRASP approach scales quite well as the problem size increases.

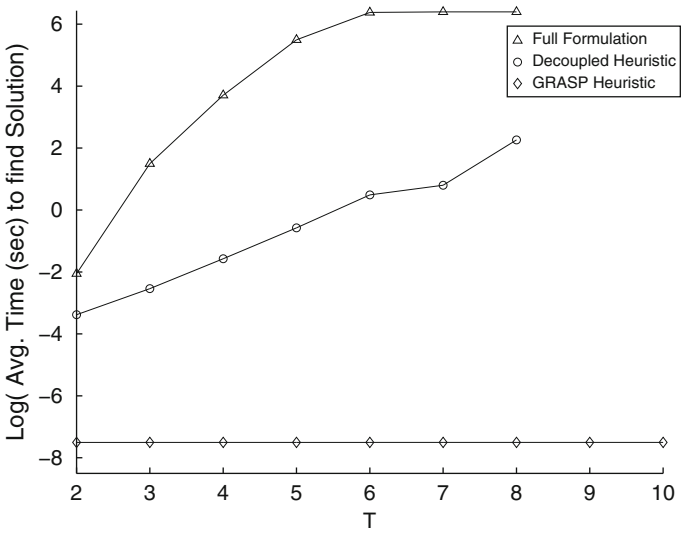

Fig. 10 Log of the average time to find a solution for $\mathcal{P} = 8$

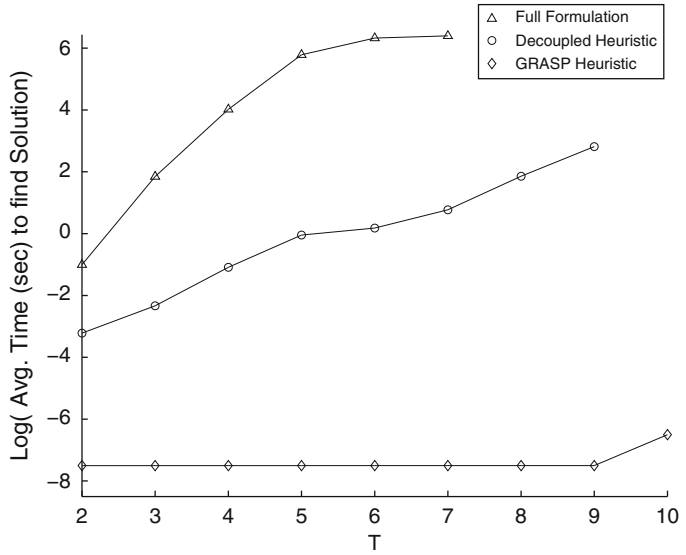

Fig. 11 Log of the average time to find a solution for $\mathcal{P} = 9$

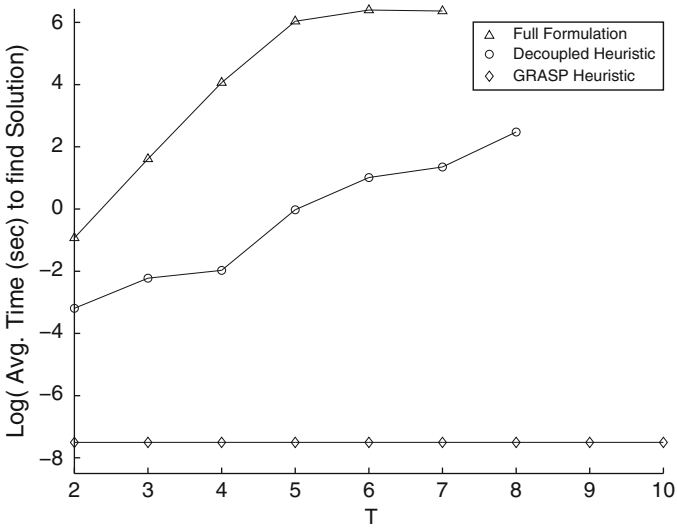

Fig. 12 Log of the average time to find a solution for $\mathcal{P} = 10$

Table 4 For the full formulation, percentage of time CPLEX (a) runs out of memory reading in the problem instance or (b)is not able to find a feasible solution in the 600 seconds

$\mathcal{T} \setminus \mathcal{P}$		2	3	4	5	6	7	8	9	10
2	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
3	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
4	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
5	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	20	10
6	(a)	0	0	0	0	0	0	0	10	90
	(b)	0	0	0	0	0	20	50	20	0
7	(a)	0	0	0	0	10	30	30	40	20
	(b)	0	0	0	0	20	70	50	60	70
8	(a)	0	0	10	20	50	100	70	100	100
	(b)	0	10	30	20	40	0	30	0	0
9	(a)	0	0	100	100	100	100	100	100	100
	(b)	10	20	0	0	0	0	0	0	0
10	(a)	10	0	100	100	100	100	100	100	100
	(b)	40	70	0	0	0	0	0	0	0

Table 5 For the decoupled heuristic, percentage of time CPLEX (a) runs out of memory reading in the problem instance or (b) is not able to find a feasible solution in the 600 seconds

$\mathcal{T} \setminus \mathcal{P}$		2	3	4	5	6	7	8	9	10
2	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
3	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
4	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
5	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
6	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
7	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
8	(a)	0	0	0	0	0	0	0	0	0
	(b)	0	0	0	0	0	0	0	0	0
9	(a)	0	0	0	0	0	0	100	0	100
	(b)	0	0	0	0	0	0	0	0	0
10	(a)	0	0	0	100	100	100	100	100	100
	(b)	0	0	0	0	0	0	0	0	0

5 Concluding Remarks

In this research we have developed a rigorous nonlinear mathematical programming formulation for workflow optimization. While this formulation has been linearized, it is still an `NP-complete` problem. Two heuristics were developed to attempt to find near-optimal solutions efficiently. The decomposition heuristic decouples the assignment of tasks to users from creating the schedule and information flow for each user. The second heuristic utilized the construction procedure of the well-known GRASP approach. We performed computational experiments for a number of combinations of users and tasks, and the results clearly show that the GRASP heuristic performs much more quickly than solving the full linearized formulation or the decomposition approach using CPLEX. In addition, the solution quality is not seen to be degraded by the GRASP heuristic. Future research will consider the case where bandwidth is not considered infinite and instantaneous, but each user will have a finite up-link and down-link capability per unit time. It is also our goal to apply this approach to real-world data (e.g., for a distributed fusion system where different nodes in a fusion network are able to combine certain pieces of information together, and the system as a whole builds up situation awareness of the domain of interest).

References

1. ILOG CPLEX: http://www.ilog.com/products/cplex, Accessed October 2011
2. Dewan, R., Seidmann, A., Walter, Z.: Workflow optimization through task redesign in business information processes. In Proceedings of the Thirty-First Hawaii International Conference on System Sciences, vol. 1, pp. 240–252, 1998
3. Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. **8**, 67–71 (1989)
4. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. J. Global Optim. **6**, 109–133 (1995)
5. Festa, P., Resende, M.G.C.: GRASP: An annotated bibliography. In: Ribeiro, C.C., Hansen, P. (eds.) Essays and Surveys in Metaheuristics, pp. 325–367. Kluwer Academic Publishers, Dordrecht (2002)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
7. Georgakopoulos, D., Hornick, M., Sheth, A.: An overview of workflow management: From process modeling to workflow automation infrastructure. Distr. Parallel Databases **3**, 119–153 (1995)
8. Joshi, A.: Reactive scheduling in workflow management systems: A branch-and-price approach. Master's thesis, University at Buffalo, Buffalo, NY, 2003
9. Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific workflow management and the kepler system. Concurrency Comput. Pract. Ex. **18**, 1039–1065 (2006)
10. Nukala, P.: Open source workflow management system with a task scheduling tool. Master's thesis, University at Buffalo, Buffalo, NY, 2006
11. Prodan, R., Fahringer, Th.: Dynamic scheduling of scientific workflow applications on the grid: a case study. In: 20th Symposium of Applied Computing (SAC 2005), pp. 687–694. ACM Press, 2005
12. Resende, M.G.C., Ribeiro, C.C.: Greedy randomized adaptive search procedures. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics, pp. 219–249. Kluwer Academic Publishers, Dordrecht (2003)
13. Tao, Q., Chang, H., Yi, Y., Gu, C., Yu, Y.: Qos constrained grid workflow scheduling optimization based on a novel pso algorithm. In: Eighth International Conference on Grid and Cooperative Computing, pp. 153–159, 2009
14. Xianwen, H., Yu, D., Bin, Z., Tingwei, C.: Reduced task-resource assignment graph based static scheduling for grid workflow application. In: 3rd International Conference on Communication Systems Software and Middleware and Workshops, pp. 736–743, 2008
15. Xiao, Z., Ming, Z., Yin, J.: Optimization of workflow pre-scheduling based on nested genetic algorithm. In: Proceedings of the 2010 Second International Conference on MultiMedia and Information Technology, vol. 1, pp. 294–297. IEEE Computer Society, 2010
16. Yu, Z., Shi, W.: An adaptive rescheduling strategy for grid workflow applications. In: 21th International Parallel and Distributed Processing Symposium (IPDPS 2007), pp. 1–8. IEEE, 2007
17. Zhang, C., Chang, R.N., Perng, C., So, E., Tang, C., Tao, T.: Qos-aware optimization of composite-service fulfillment policy. In: Proceedings of the IEEE Conference on Services Computing, pp. 1–9, 2007
18. Zhang, S., Gu, N., Li, S.: Grid workflow based on dynamic modeling and scheduling. In: International Conference on Information Technology: Coding and Computing, vol. 2, pp. 35–39, 2004

Characterization of the Operator Cognitive State Using Response Times During Semiautonomous Weapon Task Assignment

Pia Berg-Yuen, Pavlo Krokhmal, Robert Murphey, and Alla Kammerdiner

Abstract The increase in autonomy of unmanned systems is projected to continue in the foreseeable future. As a result, a single operator may be expected to monitor and control several unmanned systems. The US Air Force's Warfighter Interface Division conducted a simulation and testing with human in the loop. Their objective was to examine target acquisition performance for unaided human operators with that of an automated cooperative controller in accomplishing a complex task involving the prosecution of ground-based targets with wide area search munitions (WASMs). Their experiments provided empirical data on a humans ability to manage multiple tasks with varying mental task difficulty. The concept of the response time (RT) is often used by psychologists and neuroscientists to better understand cognitive and corresponding neural processes. Recently, the RTs have been shown to be correlated with the cognitive task difficulty. Based on these findings, a new approach for assigning task difficulty during multitasking experiments is introduced. The approach constructs a monotonously increasing mapping of the operator's RT data into the corresponding task difficulty levels with a continuous range of values.

Keywords Operator cognitive state • Response times • Weapon-task assignment

A. Kammerdiner (✉)
New Mexico State University, MSC 4230, Las Cruces, NM 88003-8001, USA
e-mail: alla@nmsu.edu

P.B.-Yuen • R. Murphey
Air Force Research Lab, Munitions Directorate, 101 West Eglin Bvld, Eglin AFB, FL 32542, USA
e-mail: pia.berg-yuen.ctr@eglin.af.mil; robert.murphey@eglin.af.mil

P. Krokhmal
Department of Mechanical and Industrial Engineering, University of Iowa, 3131 Seamans Center, Iowa City, IA 52242, USA
e-mail: krokhmal@engineering.uiowa.edu

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_4,

1 Introduction

During the past decade US military operations have seen rapid increase in the use of unmanned aerial vehicles (UAVs) [12]. In fact, the application of UAV technology has a potential to radically transform both warfare and the civilian sector. Progress in computing capabilities, artificial intelligence, secure and efficient wireless communications, materials, nanotechnology, sensors, and robotics lays foundation for new generations of UAVs. Future unmanned aircraft should have advanced capabilities that would allow it to dynamically cooperate with other UAVs, interact with humans, and act autonomously. Taking into account current technological trends in the areas of computing, robotics, and sensors, it is reasonable to expect that the autonomy of the future unmanned systems will continue to grow rapidly.

These future systems may require a single operator to monitor and control several unmanned systems simultaneously [12]. This likely trend would lead to a significant increase in the difficulty of cognitive tasks faced by system operators. Naturally, the advances in UAV capabilities would allow them to supply additional, possibly more complex information to the human operator, therefore, potentially increasing operator's cognitive load.

This raises concerns regarding the cognitive limits of a human operator, and how these limits may become a bottleneck during cooperative missions. Although the human operator in the loop is an integral part of the cooperative UAV system and should be able to impact the autonomy of UAVs, human cognitive processing has its limitations both in speed and the amount of information. During critical missions, decisions based on hastily or partially processed information by human operators may have considerable negative impact. Therefore, managing the operator's cognitive load along with the RT becomes particularly important.

In experimental research, operator's cognitive load is usually represented by a discrete quantity that typically corresponds to some level (e.g., easy, medium, hard) of cognitive task difficulty. This quantity is often assigned based on specific experimental design. For instance, the greater the number of UAVs managed by an operator during a trial, the higher the task difficulty. Although easy to implement, this approach to quantifying the cognitive load naturally has a disadvantage, as it is not sensitive to dynamic changes in actual cognitive load experienced during progression of an individual trial. When experiments are very well controlled, this may not be significant, but for more complex experiments that more closely reflect real-life missions, such simplistic assignment of the cognitive load values appears to be ill-suited. In the latter case, cognitive load may be better modeled by a bounded real-valued function of time rather than an integer constant.

This chapter presents an approach for overcoming this challenge. We propose a new way to quantify cognitive load in the cognitive tasks that are characterized by dynamically changing level of difficulty. The approach is motivated by recent research in cognitive psychology that shows that human RTs are strongly correlated with the cognitive task difficulty. These findings allow us to use operator's RTs to estimate the cognitive load of each individual operator at given time intervals based on experimental data from the simulation of real-life weapon assignment task.

The chapter is organized as follows. The following section briefly reviews recent methods used to study operator functional state (OFS) from physiological data, such as electroencephalogram (EEG) and electrooculogram (EOG). Section 3 discusses the RT as a widely used characteristic of cognitive processing, and explains its connection to the cognitive task difficulty. Additionally, the probability distributions commonly used to fit RTs in cognitive psychology and neuroscience are summarized. In Sect. 4 the new approach, which utilizes RTs to model cognitive load as a function for each individual operator based on the operator's performance during experimental trials, is described. The experiments performed on the human operators managing WASMs conducted by the air force research laboratory (AFRL) are described in Sect. 5. The results from the data analysis and discussion of the performed modeling and estimation of cognitive load for different operators are presented in Sect. 6. Finally, Sect. 7 concludes the chapter.

2 Operator Functional State

OFS is commonly defined as the immediate capacity of an operator for answering the demands of a current task or tasks using his or her cognitive and physiological resources [2]. Usually OFS is identified with cognitive load experienced by the operator.

2.1 Current Approaches for Studying OFS

In general, approaches to quantifying OFS are indirect and often based on some obvious measures of observed task performance. When performance of the task is degraded, a lower, suboptimal OFS is assumed [17]. Alternative approaches to measuring OFS indirectly are based on a subjective estimate of the mental load by an operator. It is derived from the feedback received from an operator during the task.

More direct methods for estimating OFS utilize information from physiological data that reflects changes in brain activity, heart beat, and eye movement in response to alterations in cognitive workload. Different data mining and pattern recognition procedures have been applied in direct estimation of OFS [19]. The operator's cognitive load is often classified into one of several possible categories from the collected physiological data. Artificial neural networks (ANN) and discriminant analysis are some of the most commonly used approaches.

Traditionally, research on the operator's cognitive load was focused only on the off-line analysis of the collected data. More recently, researchers became interested in solving the more practical problem of online detection of temporal changes in the cognitive state of an individual operator as a response to changes in a level of task difficulty based on real-time physiological data [3]. In the last couple of

years, several methods for online OFS estimation and change detection have been proposed [1–3]. The methods include cognitive state indicator (CSI), techniques based on control charts, and independent component analysis (ICA).

The CSI is a function that projects the multidimensional EEG/EOG signals onto the interval [0; 1] by maximizing the Kullback–Leibler distance between distributions of the signals. CSI is a continuous function of cognitive load. The CSI measure of the operator's mental workload was introduced in [2].

Control charts are analysis and visualization techniques employed in statistical process control for monitoring industrial and manufacturing processes. Techniques based on control charts are widely used in industrial engineering process control to detect "unusual" or unexpected variation. Recently these techniques have been successfully applied to aid in detecting significant changes in EEG data as a result of the driver's drowsiness [10]. Lately, the procedures for online detection of temporal changes in OFS, which are based on univariate and multivariate control charts, have been investigated [3].

Independent component analysis is a method used to extract the underlying components of signals [1]. ICA assumes that physical processes (like those in the human brain) consist of distinct operators (e.g., individual neurons), which emit signals independent of each other. When these signals are recorded by sensors, they become "mixed" and indistinguishable. ICA is a method for separating these mixed signals into their underlying components.

2.2 Task Difficulty Assignment in OFS Experiments

Procedures for online detection of OFS changes, such as the CSI, are trained to discriminate between the states with different cognitive loads. A cognitive load essentially represents a difficulty level of the assigned mental task. In simple experiments, the conditions during each trial are set up in such a way that one can easily distinguish among several discrete difficulty levels. In practice, it would make more sense to allow cognitive load to attain values from some continuous range, for instance, [0; 1], where 0 denotes the simplest mental task and 1 represents an extremely difficult task requiring maximum use of operator's cognitive resources.

Due to its dynamic nature, the problem of assigning a suitable task difficulty value in complex human experiments is not as straightforward as it is modeled in simple experiments. When simple mental tasks are performed, we normally can assume that cognitive load is approximately constant throughout such simple tasks. However, in real life operators may be faced with complex tasks that at different times involve varying degree of difficulty. Therefore, human multitasking experiments, which attempt to recreate actual conditions more accurately, naturally result in dynamically changing difficulty levels of the mental workloads faced by an operator. This makes suitable assignment of temporally changing difficulty levels quite challenging.

Our goal is to provide some answers on how this challenge can be adequately addressed. A new approached is proposed for assigning task difficulty, which

accounts for temporal changes in cognitive load. In Sects. 3 and 4, the reasons why RT can aid in the assessment of task difficulty are discussed, and the proper quantification of the difficulty of a current mental task in multitasking experiments is addressed.

3 Response Time

Estimating cognitive load is not an easy task. We propose to use RT as a proxy for cognitive load. In fact, there is evidence from psychological studies that harder cognitive tasks are likely to require additional time to complete them [11, 15]. Theoretical considerations based on RT modeling also support this observation [16].

Psychological experiments show that RT serves as a measure of the efficiency of mental processing [8,9]. For example, during a lexical decision task, high-frequency words (e.g., *to plan*) are recognized and categorized faster in comparison to low-frequency words (e.g., *to contemplate*). This substantiates the statement that a brain of an average adult person processes high-frequency words more efficiently than low-frequency words in lexical decision making [15].

RT offers researchers several important advantages. It is descriptive, as it provides quantitative information regarding mental processes. It is also convenient, since it is easy to measure in the experiments and it can be applied in a wide range of mental tasks. As a result, RT has become one of the most widely used dependent variables in psychological research [14, 16].

Historically, the majority of psychological studies used to focus their attention on analysis of mean RT across experimental conditions. Lately, there has been a major shift in the treatment of RT, which has led to increased awareness of the need to study the entire RT distribution and not just its mean. Specifically, the usefulness of considering the entire RT distribution for testing formal models of cognition is now widely acknowledged [16]. In mathematical psychology, several models have been extended to make detailed predictions regarding the shape of the entire RT distribution.

Researchers agree [11, 15, 16] that RT distributions are characterized by three important properties as evident from both careful theoretical examination and the results of the experimental studies. First, since RT cannot be negative, RT distributions are almost always skewed to the right, and therefore, clearly non-Gaussian distributions. Second, the skewness is generally more pronounced for harder cognitive tasks. Third, RT distributions are characterized by direct correlation between the mean and the variance of the RT distribution [7, 16]. In other words, an RT distribution with a greater value of the expectation is also presumed to have a larger variance.

Both parametric and nonparametric techniques are used to estimate RT distributions. Nonparametric procedures include the fixed-width histograms, quantile-based histograms, and Gaussian kernel estimators [14]. Maximum likelihood estimation and the method of least squares are two parametric approaches for fitting probability distributions to data that are often utilized for RT. An alternative procedure for

parametric estimation of RT distribution is quantile maximum likelihood (QML) estimation, which is shown to be more efficient and less biased for a number of RT models, such as the widely used ex-Gaussian [6].

A wide variety of probability models have been applied to fit observed RT data. The following six distributions are commonly used to model RT: Ratcliff's diffusion model, the ex-Gaussian distribution, the Gamma probability density function (PDF), the Poisson race model, the Wald distribution, and the Weibull density function [14]. Furthermore, the lognormal is another distribution that has been shown to be a suitable fit for RT [6, 7, 9, 13].

4 The Approach

Online detection of temporal changes in operator's cognitive load relies heavily on the analysis of the experimental data collected from operators. In fact, the procedures for online detection of critical OFS changes are constructed based on the results from data analysis of the operators' psychophysiological data. As a consequence, suitable interpretation of experimental conditions becomes vital for any analysis to be meaningful.

The difficulty level of an operator's mental task at a given point of time is one of the experimental conditions that must be determined appropriately. The level of difficulty is the parameter that is often used to construct a correspondence between the operator's physiological data and the correspondent OFS experienced during the experimental trials. As previously stated, majority of the surveyed literature indicates that the difficulty level is commonly assigned a discrete value based on an experimental setup and does not take into account individual differences. For example, the difficulty level is 0 when a number of UAVs handled by the operator during the trial is small, whereas it is assigned a 1 when this number is large. Such assignment ignores the possibility that a very experienced operator might be able to successfully handle a large number of UAVs, whereas an unexperienced operator might find a task with even a small number of UAVs difficult.

As discussed above, the assignment of mental task difficulty can be challenging. Human multitasking experiments that simulate tasks faced by operators in real-life settings have a very dynamic nature that does not fit very well into overly simplistic and static discrete assignment of a task difficulty level. It is concluded that a simple way of assigning task difficulty that takes into account only experimental conditions (e.g., number of UAVs handled during the trial) ignores temporal variations in mental task difficulty during a trial, and categorizes difficulty into a few discrete categories is not well suited for constructing procedures for an online detection of temporal changes in OFS.

Therefore, a new approach is suggested to address these challenges. The advantages of this new approach comprise: the assignment of task difficulty level based on experimental data, the capturing of cognitive abilities of each individual operator, and the incorporation of the highly dynamic nature of cognitive load

during multitasking. As discussed in Sect. 3, operator's response times for each task during every trial can be easily determined from the experimental data. Hence, using RTs, information about individual differences in each operator's cognitive abilities can be included in the determination of mental task difficulty. Furthermore, observed changes in RT from one task to the next closely reflect dynamic changes in operator's workload.

The presented approach is based on RTs, because RTs appear to be well suited to represent the amount of cognitive load. By incorporating RT into the new approach, the difficulty level is given a much wider range of values than the above-mentioned $\{0, 1\}$ simple discrete assignment. More specifically, the difficulty level becomes a function of RT onto a bounded interval. Similarly to discrete difficulty assignment, the interval can be normalized as $[0, 1]$, where 0 represents a task that requires almost no mental effort, and 1 denotes a task that needs maximum cognitive exertion.

The function that serves as an approximation of the difficulty of cognitive task from data is constructed as follows:

1. Based on recorded experimental data, compute the operator's RTs for each task in every trial.
2. For each operator, determine RT distribution that best fits the operator's RTs and estimate the distribution parameters.
3. Use the cumulative distribution function (CDF) to map each RT value into its correspondent task difficulty level on a $[0, 1]$ interval.

Note that the last two steps are used to map each response time into its respective task difficulty level. These steps are introduced to be able to compare the task difficulty among different operators. Since RT is mapped into the $[0, 1]$ interval, we could simply standardize the RT variable by applying suitable shift and scaling. However, under such transformation, the resulting $[0, 1]$ range of task difficulty would not represent uniform increase in the level of difficulty. In such a case, the increase in difficulty from l_1 to l_2 ($l_1, l_2 \in [0, 1]$) would continue to be dependent on cognitive abilities of a given operator. To be able to compare the difficulty level across different operators, one must map each operator's RT values into a common interval. This is made possible by ensuring the uniform change in difficulty level. If we apply CDF $d = F(x)$, where x represents an RT and F is an estimated CDF for a given operator, then the resulting values d will have $[0, 1]$ range and represent uniform increase in difficulty level. Thus, a common interval is obtained, which can be used to compare task difficulty among different operators.

The validity of proposed approach is based on several commonly used properties of the CDF and its inverse. Using right-continuity property of the CDF of a random variable, the pseudo-inverse of the CDF $F(x)$ is defined as

$$F^{(-1)}(x) = \inf\{x : F(x) \geq u\}, \qquad u \in (0, 1).$$

If $F(x)$ is a CDF of a continuous random variable X with bounded support, then $F(x)$ has an inverse $F^{-1}(x)$. In other words, for every $u \in (0, 1)$, there is a unique x from (a, b) (i.e., the bounded support of $F(x)$) such that $F(x) = u$. Notice that

continuous distributions with bounded support can be used to fit RT. For each operator, the RT distribution's support will lie between the minimum and maximum RTs in all trials performed by that individual.

Two other well-known properties of the CDF and its inverse, which we used here, are summarized in the following simple lemma.

Lemma 1. *Let X be a continuous random variable with cumulative distribution function (CDF) $F_X(x)$,which is strictly increasing on the possible values of X. Suppose that the inverse of $F_X(x)$ exists, and denote it by $F_X^{-1}(x)$. Let U denote a uniform random variable on $[0, 1]$. Then a random variable $F_X(X)$ is uniformly distributed on $[0, 1]$, and $F_X^{-1}(U)$ is a random variable distributed according to $F_X(x)$.*

The result in Lemma 1 ensures that the task difficulty levels are consistent across operators despite inherent variability in RTs due to individual cognitive differences among operators. This is important as it allows one to compare cognitive tasks performed by different operators on the same scale of difficulty. Notice that the assumption in Lemma 1 is consequential, as it implies that a longer RT would necessary be assigned a higher value of task difficulty using the proposed mapping approach.

In short, the constructed function for approximating cognitive task difficulty has a significant advantage. Using this function, one is able to translate the data from each individual operator into a common range of values for task difficulty level. Because the difficulty assignment is consistent across all operators, the proposed mapping can be used to compute task difficulty levels suitable for simultaneous analysis of psychophysiological data not only for a single operator, but, more importantly, among different operators.

Furthermore, the constructed function can be utilized in online algorithms for detection of temporal changes in OFS. This could be achieved by applying regression modeling to predict RT based on constant and time-dependent parameters involved in a trial, including number of UAVs, cooperative or manual mode, etc. Indeed, since during real-life multitasking, operator's RT are not known ahead of time, then one could use regression techniques to construct a predictive model of RTs using other variables (e.g., subject (or operator), number of UAVs, cooperative versus manual mode, time between the current signal and previous signal), which are known during multitasking, as the effects of the model.

5 Human Performance Experiments and Data

The US Air Force is developing advanced automation systems, such as the WASMs [18]. The WASM is a hybrid that combines the attributes of UAV such as loiter and surveillance with those of traditional fly-over-shoot-down and hit-to-kill munitions [18]. The WASM comprises semiautonomous, intelligent munitions that are able to communicate and coordinate with one another and with human operators

to accomplish a common mission by searching tactical areas of interest (TAI) and engaging ground targets encountered within the TAI [18]. Using these systems, multiple semiautonomous unmanned weapons systems could be deployed into the battle zone.

The data used in this study came from multitasking trials conducted by the AFRL Warfighter Interface Division of the Supervisory Control Interfaces Branch at Wright-Patterson AFB, OH. The trials examined target acquisition for unaided operators with that of an automated cooperative controller for a complex task involving the prosecution of ground-based targets with WASMs. The purpose of these trials was to collect empirical data to be able to assess human operator's ability to simultaneously manage multiple WASMs while performing a target search, identification, and weapon assignment task [18].

Human subjects were carefully selected to participate in the trials. The participants (or subjects) included 12 full-time civilian and military employees stationed at Wright-Patterson AFB, OH. All of them were males, and they were between 20 and 45 years old with an average age of around 30.3 years. All subjects in this sample indicated being in good to excellent health with vision correctable to 20/20, normal color vision, and normal peripheral vision. The majority of the participants reported that they had previous simulator (67%) and video game (92%) experience. Participation in the study was completely voluntary and without any compensation. Experiments were set up according to a two-by-three within-subject experimental design, and the two independent variables in this study were the number of UAVs launched (i.e., 4, 8, or 16) and the level of control for planning the attack (i.e., manual or cooperative control mode).

Computer-based simulation was used to generate specific scenarios that were presented to each participant. During each trial a new scenario was presented, and a subject was able to view the projected flight paths, the locations of the WASMs, and the targets on a display. Each subject received instructions regarding the suspected number of targets in the area to be prosecuted and the priority, high or low, of each target type. Moreover, each subject was told to attack only those items whose LADAR images matched one of the images of the assigned high and low priority targets. In other words, a subject was not suppose to attack the items that he identified as nontargets. During every trial, each subject carefully reviewed LADAR imagery from the WASM and promptly made target identification decisions (i.e., high priority target, low priority target, or nontarget). Once all the targets were identified, the subject proceeded to assign the WASMs to the targets for a subsequent attack.

6 Analytical Results

In this section, experimental data from mentioned multitasking experiment are used to illustrate how the RT distributions can be identified for a given subject. The RTs for each subject are calculated by taking the difference between recorded times from

Table 1 Log likelihood values for fitted distributions for subjects 1, 3, 5–14

	Wald	Gamma	Lognormal	Log-logistic	Normal	Weibull
Subject 1	−495.649	−736.127	−710.469	−710.713	−831.349	−748.334
Subject 3	−642.833	−914.013	−875.985	−876.722	−1,053.500	−919.817
Subject 5	−667.279	−924.617	−888.978	−893.055	−1,074.200	−923.921
Subject 6	−457.327	−700.550	−667.778	−665.630	−813.206	−712.072
Subject 7	−591.456	−839.585	−816.148	−821.849	−944.673	−847.441
Subject 8	−609.539	−869.834	−841.954	−847.138	−983.29	−875.883
Subject 9	−496.789	−754.293	−718.362	−716.462	−876.863	−765.000
Subject 10	−421.129	−698.951	−645.759	−629.068	−859.752	−716.793
Subject 11	−612.891	−891.367	−844.048	−839.616	−1,062.160	−896.610
Subject 12	−590.634	−858.134	−823.131	−825.472	−989.628	−864.084
Subject 13	−538.219	−782.799	−749.433	−749.405	−908.107	−791.817
Subject 14	−844.759	−1,079.380	−1,072.040	−1,078.800	−1,164.660	−1,082.750

the moment an item first appeared on an operator's display until the identification decision regarding that item was made by the subject. Along with the computed RTs, the preprocessed experimental data also included other variables, such as trial, subject, mode (cooperative or manual), the total number of UAVs, etc. These parameters could be used to model the subject's RT using regression.

To determine the CDF $F(x)$ for RTs of each subject, one must select a suitable distribution among several considered distributions. Besides the common RT distributions, such as Gamma, lognormal, Wald and Weibull, a number of additional PDFs, including Birnbaum–Saunders, exponential, generalized extreme value, generalized Pareto, log-logistic, and normal, were used to fit the individual response data. The method of Maximum Likelihood was used to estimate the parameters of the distribution for each subject. Table 1 contains the log likelihood values for several distributions fitted to the subjects' RT data. The Wald distribution, also known as the inverse Gaussian, appears to consistently have the largest log likelihood values across all analyzed subjects. Consequently, this distribution is chosen as the estimate of CDF $F(x)$.

The inverse Gaussian distribution arises as the first passage time distribution of Brownian motion with positive drift [4, 5]. The PDF of an inverse Gaussian distribution with parameters μ and λ is given by

$$f(t) = \sqrt{\frac{\lambda}{2\pi t^3}} \exp\left\{-\frac{\lambda(t-\mu)^2}{2\mu^2 t}\right\}, \qquad t > 0,$$

where μ is the mean and λ is a shape parameter of the distribution. Both parameters are assumed to be positive [4]. The PDF is unimodal and skewed, with a variance of μ^3/λ. Notice that μ is not a location parameter in the usual sense, since the variance of the distribution depends on μ. The CDF of the inverse Gaussian distribution is given by

Table 2 Estimated mean, variance, and parameters μ, λ of the Wald distribution for the response times of subjects 1, 3, 5–14

	$\widehat{\mu}$	Std. error	$\widehat{\lambda}$	Std. error	Mean	Variance
Subject 1	10.2179	0.5119	17.7035	1.6509	10.2179	60.2604
Subject 3	15.8202	0.9738	17.0440	1.5399	15.8202	232.3070
Subject 5	21.2376	1.7208	14.1888	1.3289	21.2376	675.1050
Subject 6	9.0873	0.4804	14.4499	1.3624	9.0873	51.9326
Subject 7	13.4156	0.7711	17.0612	1.5640	13.4156	141.5220
Subject 8	13.5650	0.8259	14.9961	1.3577	13.5650	166.4480
Subject 9	9.7068	0.5206	14.2968	1.3161	9.7068	63.9726
Subject 10	7.0183	0.3257	13.2476	1.1945	7.0183	26.0950
Subject 11	14.1012	0.8795	14.7370	1.3288	14.1012	190.2670
Subject 12	12.8186	0.7866	13.9514	1.2631	12.8186	150.9740
Subject 13	12.7031	0.7327	16.9702	1.6000	12.7031	120.7920
Subject 14	31.4256	2.1349	27.7918	2.5110	31.4256	1,116.6900

$$F(t) = \Phi\left(\sqrt{\frac{\lambda}{t}}\left(\frac{t}{\mu} - 1\right)\right) + \exp\left\{\frac{2\lambda}{\mu}\right\} \Phi\left(-\sqrt{\frac{\lambda}{t}}\left(\frac{t}{\mu} + 1\right)\right), \qquad t > 0,$$

where Φ denotes the CDF of the standard normal distribution.

The inverse Gaussian distribution was used to estimate the CDF of the individual response times for all 12 subjects who participated in the study (n.b., subjects 2 and 4 did not participate in the study). Table 2 summarizes the estimated parameters $\widehat{\mu}$ and $\widehat{\lambda}$ of the distribution, together with the estimated mean and variance values for subjects 1, 3, 5–14. The estimated mean RT values range from as small as 7.0183 for subject 10 to as large as 31.4256 for subject 14. Note that the second largest mean RT value is 21.2376 (subject 5). While the shape RT values have a narrower range between 13.2476 (subject 10) and 17.7035 (subject 1), when excluding a possible outlier of 27.7918 (subject 14). The estimated variance also varies significantly among subjects, mostly due to the variability in the parameter μ. Furthermore, subject 14 appears to have the RTs significantly different from the other subjects. It is concluded that an individual estimation of the CDF $F(x)$ for each subject is necessary.

It may be necessary to predict the subjects RTs based on the observed characteristics or parameters during the trial. Regression could be used as a predictive modeling technique. However, in many cases to be a suitable regressive model, the normality of the data is often required. Normalizing the RT data would allow one to apply regression for predicting the subjects' RTs.

Accordingly, the inverse transform and the natural logarithm were applied with a purpose of normalizing the RT data. When the data followed the inverse Gaussian distribution, the inverse mapping $f(x) = 1/x$ was applied to transform the data to near normality, whereas the natural logarithm $f(x) = \log x$ proved to be effective for normalizing lognormally distributed data.

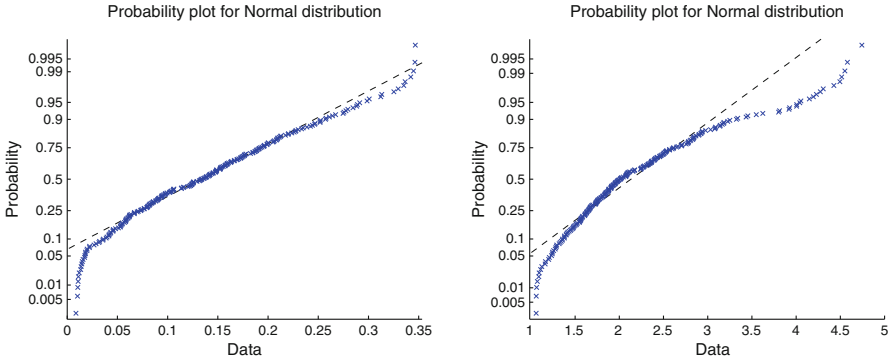

Fig. 1 Probability plots of the transformed response times for subject 11. The inverse transform (*left*) and the natural logarithm (*right*) are applied with a purpose of normalizing the data

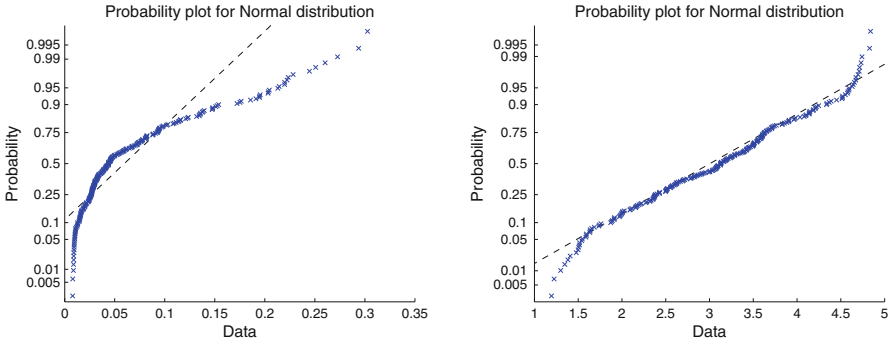

Fig. 2 Probability plots of the transformed response times for subject 14. The inverse transform (*left*) and the natural logarithm (*right*) are applied with a purpose of normalizing the data

Probability plots for the transformed RTs for subjects 11 and 14 are presented in Figs. 1 and 2, respectively. For each figure, the plot on the left was produced by applying an inverse transformation $f(x) = 1/x$ to the data, whereas the plot on the right was obtained by transforming the RT data using a natural logarithm $f(x) = \log x$.

The inverse transform and the natural logarithm produce very different effects on the subject 11 and subject 14 data. On the one hand, the plots in Fig. 1 show that despite a presence of few outliers, the inverse transform appears to normalize the RT of subject 11 better than the logarithm. On the other hand, the logarithm works better than the inverse transform for normalizing the RT of subject 14, as illustrated in Fig. 2. Likewise, probability plots of the transformed RTs were produced for the remaining ten subjects. It was found that the inverse function works better than the logarithm for transforming the RT data to near normality for all subjects with the exception of subject 14.

7 Conclusion

This chapter presented an approach aimed at addressing some of the inherent challenges in the assignment of the difficulty level in complex multitasking experiments. The developed approach constructed the difficulty level of a current cognitive task as a monotonously increasing function of the observed response times for each operator, based on the known close relationship between the cognitive load and the response time.

The new approach has several advantages in comparison to the commonly applied simplistic assignment of the task difficulty depicted in the literature. Specifically, the developed approach takes into account individual differences among the operators. It is also capable of capturing the dynamic nature of changes in the difficulty of cognitive tasks faced during multitasking. Lastly, it treats the possible levels of task difficulty as a continuous rather than discrete range of values.

Moreover, the introduced approach requires estimation of the CDF of the response times for each operator. Caution must be exercised when applying the approach to the response data having large variability. For the majority of the subjects, the response times during the weapon assignment task can be described by the inverse Gaussian distribution and can be transformed to near normality using the inverse function. However, some operators may have response times, for which the inverse Gaussian distribution and the inverse transform may not work well. Future work includes a more in-depth study of the characterization of the operator cognitive state using response times and other relevant parameters.

Acknowledgements A. Kammerdiner and P. Berg-Yuen gratefully acknowledge support from the national research council (NRC) Postdoctoral Fellowship Program.

References

1. Cannon, J.A.: Statistical analysis and algorithms for online change detection in real-time psychophysiological data, Thesis, University of Iowa (2009)
2. Cannon, J.A., Krokhmal, P.A., Lenth, R.V., Murphey, R.: An algorithm for online detection of temporal changes in operator cognitive state using real-time psychophysiological data. Biomed. Signal Process. Contr. **5**(3), 229–236 (2010)
3. Cannon, J.A., Krokhmal, P.A., Chen, Y., Murphey, R.: Detection of temporal changes in psychophysiological data using statistical process control methods. Comput. Meth. Programs Biomed. **102**(2) (2011)
4. Chhikara, R.S., Folks, L.: The inverse Gaussian distribution as a lifetime model. Technometrics **19**(4), 461–468 (1977)
5. Chhikara, R.S., Folks, L.: The Inverse Gaussian Distribution: Theory, Methodology, and Applications. Marcel Dekker, Inc. New York (1989)
6. Heathcote, A., Brown, S., Mewhort, D.J.K.: Quantile maximum likelihood estimation of response time distributions. Psychonomic Bull. Rev. **9**, 394–401 (2002)
7. Holden, J.G., Van Orden, G.C., Turvey, M.T.: Dispersion of response times reveals cognitive dynamics. Psychol. Rev. **116**(2), 318–342 (2009)

8. Link, S.W.: The Wave Theory of Difference and Similarity. Erlbaum, Hillsdale (1992)
9. Luce, R.D.: Response Times. Oxford University Press, New York (1986)
10. Murata, A., Nishijima, K.: Evaluation of drowseness by EEG analysis – Basic Study on ITS Development for the Prevention of Drowsy Driving. In: Proceedings of the 4th International Workshop on Computational Intelligence & Applications, 95–98 (2008)
11. Ratcliff, R.: A diffusion model account of response time and accuracy in a brightness discrimination task: Fitting real data and failing to fit fake but plausible data. Psychonomic Bull. Rev. **9**, 278–291 (2002)
12. Russell, M.: Unmanned Systems: Can the Industrial Base Support the Pentagon's Vision? National Defense magazine, July 2010 (2010)
13. Storms, G., Delbeke, L.: The irrelevance of distributional assumptions on reaction times in multidimensional scaling of same/different judgment tasks. Psychometrika **57**(4), 559–614 (1992)
14. Van Zandt, T.: How to fit a response time distribution. Psychonomic Bull. Rev. **7**(3), 424–465 (2000)
15. Wagenmakers, E.-J., Grasman, R.P.P.P., Molenaar, P.C.M.: On the relation between the mean and the variance of a diffusion model response time distribution, J. Math. Psychol. **49**, 195–204 (2005)
16. Wagenmakers, E.-J., Brown, S.: On the linear relation between the mean and the standard deviation of a response time distribution. Psychol. Rev., **114**(3), 830–841 (2007)
17. Wilson, G.E., Russell, C.: Operator functional state classification using neural networks with combined physiological and performance features. In: Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting, 1099–1101 (1999)
18. Warfield, L., Carretta, T.R., Patzek, M.J., Estepp, J.R., O'Neal, J.K.: Comparing Manual and Cooperative Control Mission Management Methods for Wide Area Search Munitions, AFRL-RH-WP-TR-2009-0021, Final Report, AFOSR (2009)
19. Wilson, G.E., Russell, C.: Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding. Hum. Factors **49**(6), 1005–1018 (2007)

Correntropy in Data Classification

Mujahid N. Syed, Jose C. Principe, and Panos M. Pardalos

Abstract In this chapter, the usability of the correntropy-based similarity measure in the paradigm of statistical data classification is addressed. The basic theme of the chapter is to compare the performance of the correntropic loss function with the conventional quadratic loss function. Moreover, the issues related to the non-convexity of the correntropic loss function are considered while proposing new classification methods. The proposed methods incorporate the correntropic loss function via the notions of convolution smoothing and simulated annealing optimization algorithms. Two nonparametric classification methods based on the correntropic loss function are proposed and compared with the conventional parametric and nonparametric methods. Specifically, the classification performance of the proposed artificial neural network-based methods are not only compared with their conventional counterparts but also with the kernel-based soft margin support vector machines. Experimental studies with Monte Carlo-based simulations show the validity of the proposed methods in the data classification.

Keywords Statistical classification • Correntropy • Convolution smoothing • Simulated annealing • Artificial neural networks • Support vector machines

M.N. Syed (✉)
University of Florida, Gainesville, FL, USA
e-mail: smujahid@ufl.edu

J.C. Principe
Computational NeuroEngineering Laboratory, University of Florida, Gainesville, FL, USA
e-mail: principe@cnel.ufl.edu

P.M. Pardalos
Center of Applied Optimization, University of Florida, Gainesville, FL, USA
e-mail: pardalos@ufl.edu

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_5,

1 Introduction

The theme of this chapter is to present a comparative study on performance behavior of the classical quadratic loss function and the proposed correntropic loss function under the context of statistical data classification. The proposed loss function under a nonparametric framework is compared with conventional parametric and nonparametric approaches. The motivation to compare the two loss functions (quadratic and correntropic) is due to the reasons mentioned in the following paragraph.

The quadratic loss function measures data and noise on the same norm scale, i.e., it does not differentiate noise from data. Thereby, the obtained classification rule may also reflect the effect of noisy samples (outliers), in the case of a noisy data. However, the quadratic function is still widely used in classification, and its usability is attributed to its convex nature. Due to convexity, efficient learning algorithms can be designed based on the gradient descent method and similar greedy methods. When compared to quadratic loss function, on the other hand, the correntropic function does not measure noise and data on the same norm scale. Based on the kernel width of the correntropic function, noise can be winnowed from the data. Such kernel width can be defined a priori, when the structure of data set is well understood. In addition to that, depending upon the kernel width, the correntropic function changes its nature from a convex function to a non-convex function. Hence, it restricts the use of gradient descent methods in proposing learning algorithms. In general, injecting non-convexity to the problem is undesirable, and often raises concerns about the added difficulty in the solution search. However, there are two crucial points that should be highlighted for the use of the correntropic function. The first and foremost, the correntropic function is not always non-convex, and at certain values of kernel width the correntropic loss function behaves exactly as a quadratic loss function (quadratic loss function can be considered as a specific case of correntropic loss function when the error is bounded). The next important point to highlight is the inevitability of noise in the data. With the development of so many noise filters, still, it is impossible to completely de-noise the real-world data. Under the reference of the above stated points, the idea of using the correntropic function instead of the quadratic function in data classification can be fruitful.

Thus, the comparative study on the performance behavior of the loss functions aptly forms the theme of our chapter. Furthermore, nonparametric approaches like artificial neural networks perform empirical risk minimization (ERM), whereas, parametric approaches like support vector machines perform structural risk minimization (SRM). Different studies in the literature do not consider “noise” aspect in the data while comparing the efficiency of ERM or SRM. Thus, it would be of practical importance to compare the behavior of SRM and ERM in noisy data. Therefore, in this chapter, a numerical comparison of conventional parametric and nonparametric approaches with the proposed nonparametric approach is considered. The goal of this chapter is not to conclude the superiority of parametric or

nonparametric approach, but the goal is to show the usability of correntropic loss function. In the following part of this section, the basic concepts of data analysis will be reviewed.

Data analysis is an interdisciplinary field, including statistics, database design, machine learning, and optimization. It can be defined in simple terms as *the process of extracting knowledge from a raw data set*. In general, any data analysis should involve the following five sequential steps:

- Objective
  - The first and most important step in data analysis is the objective of the analysis. It should be well defined and clear in nature. Based on the objective, the later steps are customized. Typically, the objectives may involve one or more than one of the following major criteria:
    - Regression
      Literally the term "Regression" means a return to formal or primitive state. Statistical regression involves the idea of finding an underlying primitive relationship between the causal variables and the effect variables. Moreover, the unstated basic assumption in statistical regression is that all the data belongs to a single class.
    - Classification
      Literally "Classification" means a process of classifying something based on shared characteristics. Statistical classification is a supervised learning method that involves classifying uncategorized data based on the knowledge of categorized data. The unstated basic assumption in statistical classification is that the uncategorized data should belong to any of the known data classes.
    - Clustering
      Literally "Clustering" means congregating things together based on their particular characteristics. Statistical clustering is an unsupervised learning method which aims to cluster data based on defined nearness measure. It involves multiple classes, and for each class an underlying relationship is to be found. Moreover, there is no prior knowledge about available data classes.
- Data representation
  - The *data set* comes in different forms and representations. It can represent a qualitative or quantitative data (in the form of numbers, text, patterns, or categories). Based on the objective of data analysis, a suitable data representation should be selected. A generalized way to represent data is in the form of an $n \times p$ matrix, also known as *flat representation*. The rows (records, individual, entities, cases, or objects) represent one data point, and for each data point a column (attribute, feature, variable, or field) represents a measurement value.

- Knowledge representation
  - The extracted *knowledge* can be *represented* in the form of relationships (between inputs and outputs) and/or summaries (novel ways to represent same data). The way of representing the relationships (or summaries) depends upon the field of research, and the final audience (i.e., it should be novel, but more importantly understandable to the reader). The relationships/summaries (often referred to as models or patterns) can be represented but not limited to following forms: Linear equations, Graphs, Trees, Rules, Clusters, Recurrent patterns, Neural networks, etcetera. Typically, the type of representation for relationships/summaries should be selected a priori before analyzing the data.
- Loss function
  - The *loss function* is a measure function that accounts for the error between the predicted output and actual output. The selection or design of loss function depends upon two main criteria. At first, it should appropriately reflect the error between the predicted output and actual output. Next, the loss function should be easily incorporable with an optimization algorithm. In addition to that, given an instance of predicted output and actual output, the loss function should give the error value in polynomial time. The longer it takes to calculate the error, the lesser is the efficiency of the optimization algorithm. There are two main classical loss functions, namely: absolute error, mean square error. Typically, the mean square error (commonly known as quadratic loss function) is used often as a loss function.
- Optimization algorithm
  - The knowledge representation, selected a priori, is trained (using an *optimization algorithm*) on the data set to minimize the loss function. Thus, this assures that the represented knowledge aptly imitates the real system (the source or generator of the data set). Such training algorithm, also known as learning algorithm, is based on some optimization methods. Classically, a parametric representation is encouraged, and is accompanied by an exact optimization method. Although a parametric representation requires in-depth knowledge of the given data set they were given superiority over nonparametric methods due to the existence of efficient exact optimization methods. Moreover, exact solution methods are suitable for a limited class of parametric representations, thus they limit the scope of knowledge representation. Recent developments in the use of nonparametric methods like artificial neural networks have widened the scope of knowledge representation. However, due to the use of exact methods, they have not been utilized to their full potential. Lately, due to the development in heuristic optimization methods, the use of nonparametric methods has become desirable, and enlarged the scope of knowledge representation.

In this chapter, the objective for data analysis will be statistical classification. Furthermore, data in the form of an $n \times p$ matrix (flat representation) will be used

for the analysis. Although an $n \times p$ matrix representation is not the only way of representation, it will be used as a basic data representation scheme for the rest of the chapter. When it comes to the knowledge representation, both parametric and nonparametric representations will be considered. Moreover, as stated at the beginning, a comparative analysis of the two types of loss functions is the theme of the chapter. In addition to that, both exact and heuristic methods will be used as the optimization algorithms.

The rest of the chapter is organized as follows. In Sect. 2 all the required preliminary topics will be reviewed. Section 3 will present different classification methods that will be used in this chapter. Section 4 will illustrate the use of proposed methods to real-world data (simulations and results are presented). Section 5 concludes the chapter.

2 Methods

In this section, all the major preliminary topics that will be required to understand the proposed methods will be discussed. The purpose of reviewing these topics is to provide a sufficient background information to a novice reader. However, they are by no means serve as a comprehensive discussion, and interested readers will be directed to the appropriate references for the detailed discussions. Following are the topics that will be presented in this section:

- Classification.
- Correntropic function.
- Convolution smoothing (CS).
- Simulated annealing (SA).
- Artificial neural network (ANN).
- Support vector machine (SVM).

2.1 Classification

Classification (strictly speaking, statistical classification) is a supervised learning methodology of identifying (or assigning) class labels to an unlabeled data set (a subpopulation of data, whose class is unknown) from the knowledge of a pre-labeled data set (another subpopulation of the same data, whose class is known). The knowledge of pre-labeled data set can be used to generate an optimal rule, based on the theory of learning [34, 35]. More specifically, the optimal rule (discriminant function f) is generated in such a way that it will minimize the risk of assigning incorrect class labels [2, 18]. The classification problem will be defined in the following paragraph.

Let D_n represent the data set containing the observations, defined as $D_n = \{(\mathbf{x}_i, y_i), i = 1, .., n : \mathbf{x}_i \in \mathbb{R}^m \wedge y_i \in \{-1, 1\}\}$, where $\mathbf{x}_i$ is an input vector, and y_i is the class label for the input vector. Under the assumption that $(\mathbf{x}_i, y_i)$ is an independent and identical realization of random pair $(\mathbf{X}, Y)$ classification problem can be defined as: to find a function f from a class of functions Γ, such that f minimizes the risk, $R(f)$. Thus, classification problem can be written as

minimize:

$$R(f) \tag{1a}$$

subject to:

$$(\mathbf{x}_i, y_i) \in D_n \quad \forall\, i = 1, \ldots, n, \tag{1b}$$

$$\mathbf{x}_i \in \mathbb{R}^m \quad \forall\, i = 1, \ldots, n, \tag{1c}$$

$$y_i \in \{-1, 1\} \quad \forall\, i = 1, \ldots, n, \tag{1d}$$

$$f \in \Gamma, \tag{1e}$$

where $R(f)$ is defined as

$$\begin{aligned} R(f) &= P(Y \neq sign(f(\mathbf{X})), \\ &= E[l_{0-1}(f(\mathbf{X}), Y)], \end{aligned} \tag{2}$$

where *sign* is the $signum$ function, and l_{0-1} is the zero one loss function, they are defined as

$$sign(f(\mathbf{X})) = \begin{cases} +1 & \text{if } f(\mathbf{X}) > 0, \\ -1 & \text{if } f(\mathbf{X}) < 0, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

$$l_{0-1}(f(\mathbf{x}), y) = ||(-yf(\mathbf{x}))_+||_0, \tag{4}$$

where $(.)_+$ denotes the positive part and $||.||_0$ denotes the L_0 norm. When $f(\mathbf{x}) = 0$, the above definition does not reflect error, however, this is a rare case and can be easily avoided or adjusted (i.e., by considering $||(f(\mathbf{x}) - y)_+||_0$). Moreover, it is clear from the definition of $R(f)$ that it requires the knowledge of $P(\mathbf{X}, Y)$, the joint probability distribution of the random pair $(\mathbf{X}, Y)$. Usually, the joint distribution is unknown. This leads to the calculation of empirical risk function $\widehat{R}(f)$, which is given as

$$\widehat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} l_{0-1}(y_i f(\mathbf{x}_i)). \tag{5}$$

At this juncture, only ERM is considered, and any discussion pertaining to SRM is avoided. However, SRM will be discussed when the notion of support vector machine is presented. In addition to that, it is not easy to find the optimal solution $f^\star$ of problem stated in formulation 1, since the space of functions class Γ is huge, and there is no efficient way to search over such space. In order to find the solution, a usual approach is to select the class of functions a priori, and then try to find the best function from the selected function class $\widehat{\Gamma}$. Generally, the selected class of functions can be categorized as parametric (specific) or nonparametric (general) class. Based on the category of the function class, different learning algorithms can be used to minimize the loss function. Therefore, with the above-stated restrictions, the classification problem can be represented as

minimize:

$$\widehat{R}(f) \tag{6a}$$

subject to:

$$(\mathbf{x}_i, y_i) \in D_n \quad \forall\, i = 1, \ldots, n, \tag{6b}$$

$$\mathbf{x}_i \in \mathbb{R}^m \quad \forall\, i = 1, \ldots, n, \tag{6c}$$

$$y_i \in \{-1, 1\} \quad \forall\, i = 1, \ldots, n, \tag{6d}$$

$$f \in \widehat{\Gamma}. \tag{6e}$$

In summary, usually, both $\widehat{R}(f)$ and $\widehat{\Gamma}$ will be selected before finding $f^\star$. Moreover, the type of risk function and the function class selected will significantly determine the accuracy of classification method. In the following section, a new risk function that can be used in data classification is proposed.

2.2 Correntropic Function

Although the classification problem stated in (6) looks simple, it has an inherent difficulty, due to the nonconvex and noncontinuous loss function defined in (4). Furthermore, the search over the $\widehat{\Gamma}$ function space is another difficulty in solving problem (6). In this section, the goal is to propose a loss function that can efficiently replace the loss function given in (4). Conventionally, the zero one loss function is replaced by a quadratic loss function, i.e., the quadratic risk is given as

$$\begin{aligned} R(f) &= E[(Y - f(\mathbf{X})^2], \\ &= E[(\varepsilon)^2]. \end{aligned} \tag{7}$$

In general, the knowledge of probability distribution function (PDF) of ε is required to calculate the above risk function. However, the quadratic risk can be approximated by the following empirical quadratic risk function:

$$\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2\,, \tag{8}$$

where n is the number of samples. The replacement of 0–1 loss function with quadratic loss function makes problem (6) computationally easy to solve (due to its convex nature). Moreover, if the function class $\widehat{\Gamma}$ is smooth, then the problem can be solved by gradient descent methods. However, the quadratic loss function performs poorly in noisy data i.e., the computational simplicity has its price in the classification performance. Hence, usual gradient descent based optimization methods with quadratic loss function may not provide the global optimal solution for the class of functions selected ($\widehat{\Gamma}$). In order to overcome this difficulty, the use of correntropic loss function is addressed.

Correntropy (strictly speaking, should be called as cross correntropy) is a generalized similarity measure between any two arbitrary random variables (X, Y). It is defined as [27]

$$v(X, Y) = E_{XY}\left[k_\sigma(X - Y)\right], \tag{9}$$

where k_σ is any form of transformation kernel function, usually taken as Gaussian Kernel. In order to define the correntropic risk function, consider a function $\phi_{\beta,\sigma}(f(\mathbf{x}), y)$[1] defined as

$$\begin{aligned}\phi_{\beta,\sigma}(f(\mathbf{x}), y) &= \beta[1 - k_\sigma(1 - yf(\mathbf{x}))],\\ &= \beta[1 - k_\sigma(1 - \alpha))],\end{aligned} \tag{10}$$

where $\alpha = yf(\mathbf{x})$ is called the margin, $\beta = [1 - e^{\left(\frac{-1}{2\sigma^2}\right)}]^{-1}$ is a positive scaling factor, and k_σ is the Gaussian kernel with width parameter σ. Using this information, the correntropic risk function can be rewritten as:

$$\begin{aligned}R(f) &= E[\phi_{\beta,\sigma}(f(\mathbf{X}), Y)],\\ &= E[\beta(1 - k_\sigma(1 - Yf(\mathbf{X})))],\\ &= \beta(1 - E[k_\sigma(1 - Yf(\mathbf{X}))]),\\ &= \beta(1 - v(1 - Yf(\mathbf{X}))]),\\ &= \beta(1 - v(Y - f(\mathbf{X}))]),\\ &= \beta(1 - v(\varepsilon)).\end{aligned} \tag{11}$$

[1]This function has its roots from correntropy function (see [22] for more details).

Due to the unavailability of PDF function, similar to quadratic loss function, the empirical correntropic risk function can be defined as

$$\widehat{R}(f) = \beta(1 - \widehat{v}(\varepsilon)), \tag{12}$$

where $\widehat{v}(\varepsilon) = \frac{1}{n}\sum_{i=1}^{n} k_\sigma(y_i - f(\mathbf{x_i}))$, n is the number of samples.

The characteristics of this function for different values of the width parameter are shown in Fig. 1.

Clearly, from Fig. 1, it can be seen that the function $\phi_{\beta,\sigma}(f(\mathbf{x}), y)$ is convex for higher values of kernel width parameter ($\sigma > 1$), and as the parameter value decreases, it becomes non-convex. For $\sigma = 1$ it approximates the hinge loss function (hinge loss function is a typical function often used in SVMs). However, for smaller values of kernel width the function almost approximates the 0–1 loss function, which is mostly an unexplored territory for typical classification problems. In fact, any value of kernel width apart from $\sigma = 2$ or 1 has not been studied for other loss functions. This peculiar property of correntropic function can be harmoniously used with the concept of convolution smoothing for finding global optimal solutions. Moreover, with a fixed lower value of kernel width, suitable global optimization algorithms (heuristics like simulated annealing) can be used to find the global optimal solution. In the following sections, elementary ideas about different optimization algorithms that can be used with the correntropic loss function are discussed.

2.3 Convolution Smoothing

A convolution smoothing (CS) approach[2] forms the basis for one of the proposed methods of correntropic risk minimization. The main idea of CS approach is sequential learning, where the algorithm starts from a high kernel width correntropic loss function and smoothly moves towards a low kernel width correntropic loss function (approximating original loss function). The suitability of this approach can be seen in [29], where the authors used a two-step approach for finding the global optimal solution. The current proposed method is a generalization of the two-step approach. Before discussing the proposed method, consider the following basic framework of CS.

A general unconstrained optimization problem is defined as

$$\text{minimize:} \quad g(u) \tag{13a}$$

$$\text{subject to:} \quad u \in \mathbb{R}^n, \tag{13b}$$

[2] A general approach for solving non-convex problems via convolution smoothing was proposed by Styblinski and Tang [30] in 1990.

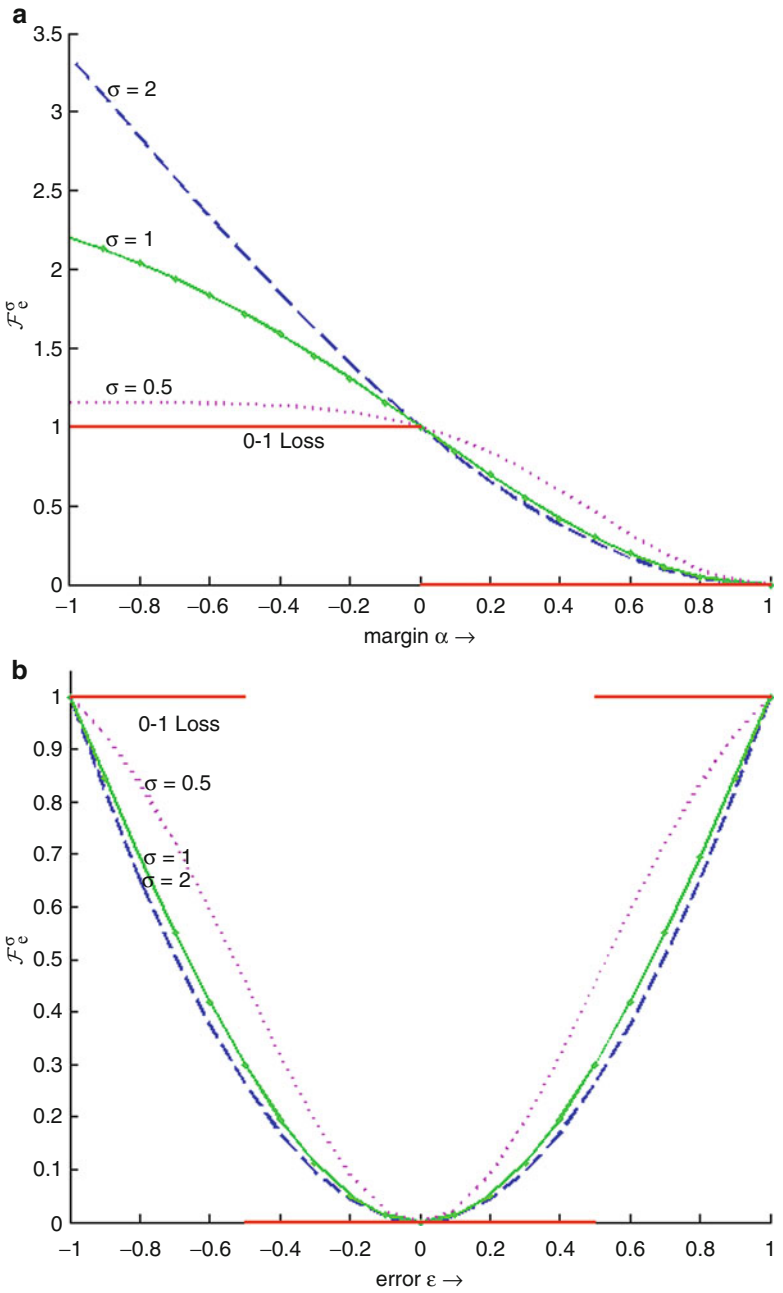

Fig. 1 Correntropic function and 0–1 loss function

where $g : \mathbb{R}^n \mapsto \mathbb{R}$. The complexity of solving such problems depends upon the nature of function g. If g is convex in nature, then a simple gradient descent method will lead to the global optimal solution. Whereas, if g is non-convex, then gradient descent algorithm will behave poorly, and converges to a local optimal solution (or in the worst case converges to a stationary point).

CS is a heuristic-based global optimization method to solve problems of type (13) when g is non-convex. This is a specialized stochastic approximation method introduced in [24]. Usage of convolution in solving convex optimization problems was first proposed in [3]. Later, as an extension, a generalized method for solving non-convex unconstrained problems is proposed in [26]. The main motivation behind CS is that the global optimal solution of a multi-extremal function g can be obtained by the information of a local optimal solution of its smoothed function. It is assumed that the function g is a convoluted function of a convex function g_0 and other non-convex functions $g_i \forall = 1, \ldots, n$. The other non-convex functions can be seen as noise added to the convex function g_0. In practice g_0 is intangible, i.e., it is impractical to obtain a deconvolution of g into g_i's, such that $\text{argmin}_u\{g(u)\} = \text{argmin}_u\{g_0(u)\}$. In order to overcome this difficulty, a smoothed approximation function $\widehat{g}$ is used. This smoothed function has the following main property:

$$\widehat{g}(u, \lambda) \longrightarrow g(u) \quad \text{as } \lambda \longrightarrow 0, \tag{14}$$

where λ is the smoothing parameter. For higher values of λ, the function is highly smooth (nearly convex), and as the value of λ tends towards zero, the function takes the shape of original non-convex function g.

Such smoothed function can be defined as

$$\widehat{g}(u, \lambda) = \int_{-\infty}^{\infty} \widehat{h}((u - v), \lambda)\ g(v)\ \mathrm{d}v, \tag{15}$$

where $\widehat{h}(v, \lambda)$ is a kernel function, with the following properties:

- $\widehat{h}(v, \lambda) \longrightarrow \delta(v)$, as $\lambda \longrightarrow 0$; where $\delta(v)$ is Dirac's delta function.
- $\widehat{h}(v, \lambda)$ is a probability distribution function.
- $\widehat{h}(v, \lambda)$ is a piecewise differentiable function with respect to u.

Moreover, the *smoothed gradient* of $\widehat{g}(u, \lambda)$ can be expressed as

$$\nabla\widehat{g}(u, \lambda) = \int_{-\infty}^{\infty} \nabla\widehat{h}(v, \lambda)\ g(u - v)\ \mathrm{d}v. \tag{16}$$

Equation (16) highlights a very important aspect of CS, it states that information of $\nabla g(v)$ is not required for obtaining the smoothed gradient. This is one of the crucial aspects of smoothed gradient that can be easily extended for non-smooth optimization problems, where $\nabla g(v)$ does not usually exist.

Furthermore, the objective of CS is to find the global optimal solution of function g. However, based on the level of smoothness, a local optimal solution of the smoothed function may not coincide with the global optimal solution of the original function. Therefore, a series of sequential optimizations are required with different level of smoothness. Usually, at first, a high value of λ is set, and an optimal solution $u_\lambda^\star$ is obtained. Taking $u_\lambda^\star$ as the starting point, the value of λ is reduced and a new optimal value in the neighborhood of $u_\lambda^\star$ is obtained. This procedure is repeated until the value of λ is reduced to zero. The idea behind these sequential optimizations is to end up in a neighborhood of $u^\star$ as $\lambda \longrightarrow 0$, i.e.,

$$u_\lambda^\star \longrightarrow u^\star \quad as\ \lambda \longrightarrow 0, \tag{17}$$

where $u^\star = \text{argmin}\{g(u)\}$. The crucial part in the CS approach is the decrement of the smoothing parameter. Different algorithms can be devised to decrement the smoothing parameter. In [30] a heuristic method (similar to simulated annealing) is proposed to decrease the smoothing parameter.

Apparently, the main difficulty of using the CS method to any optimization problem is defining a smoothed function with the property given by (14). However, the CS can be used efficiently with the proposed correntropic loss function, as the correntropic loss function can be seen as a generalized smoothed function for the true loss function (see Fig. 1). The kernel width of correntropic loss function can be visualized as the smoothing parameter.

Therefore, the CS method is applicable in solving the classification problem, when suitable kernel width is unknown a priori (a practical situation). On the other hand, if appropriate value of kernel is width known a priori (maybe an impractical assumption, but quiet possible), then other efficient methods may be developed. If the known value of kernel width in the correntropic loss function results into a convex function, then any gradient descent based method can be used. However, when the resulting correntropic loss function is non-convex, then global optimization approaches should be used. Specifically, for such cases (when the correntropic loss function results in a non-convex function) the use of simulated annealing is proposed. In the following section, the basic description of simulated annealing is presented.

2.4 Simulated Annealing

Simulated annealing (SA) is a meta-heuristic method which is employed to find a good solution to an optimization problem. This method stems from thermal annealing which aims to obtain a perfect crystalline structure (lowest energy state possible) by a slow temperature reduction. Metropolis et al. in 1953 simulated this processes of material cooling [6], Kirkpatrick et al. applied the simulation method for solving optimization problems [13, 20].

Simulated annealing can be viewed as an upgraded version of greedy neighborhood search. In neighborhood search method, a neighborhood structure is defined in the solution space, and the neighborhood of a current solution is searched for a better solution. The main disadvantage of this type of search is its tendency to converge to a local optimal solution. SA tackles this drawback by using concepts from Hill-climbing methods [17]. In SA, any neighborhood solution of the current solution is evaluated and accepted with a probability. If the new solution is better than the current solution, then it will replace the current solution with probability 1. Whereas, if the new solution is worse than the current solution, then the acceptance probability depends upon the control parameters (temperature and change in energy). During the early iterations of the algorithm, temperature is kept high, and this results in a high probability of accepting worse new solutions. After a predetermined number of iterations, the temperature is reduced strategically, and thus the probability of accepting a new worse solution is reduced. These iterations will continue until any of the termination criteria is met. The use of high temperature at the earlier iterations (low temperature at the later iterations) can be viewed as exploration (exploitation, respectively) of the feasible solution space. As each new solution is accepted with a probability it is also known as stochastic method. A complete treatment of SA and its applications is carried out in [23]. Neighborhood selection strategies are discussed in [1]. Convergence criteria of SA are presented in [14].

In this work, SA will be used to train the correntropic loss function when the information of kernel width is known a priori. Although the assumption of known kernel width seems implausible, any known information of an unknown variable will increase the efficiency of solving an optimization problem. Moreover, a comprehensive knowledge of data may provide the appropriate kernel width that can be used in the loss function. Nevertheless, when kernel width in unknown, a grid search can be performed on the kernel width space to obtain appropriate kernel width that maximizes the classification accuracy (this is a typical approach while using kernel-based soft margin SVMs, which generally involves grid search over a two dimensional parameter space).

So far, no discussion about the function class ($\widehat{\Gamma}$) is addressed. In the current work, a nonparametric function class namely artificial neural networks, and a parametric function class namely support vector machines is considered. In the following section, an introductory review of artificial neural networks is presented.

2.5 Artificial Neural Networks

Curiosity of studying the human brain led to the development of ANNs. Henceforth, ANNs are the mathematical models that share some of the properties of brain functions, such as nonlinearity, adaptability, and distributed computations. The first mathematical model that depicted a working ANN used the perceptron, proposed by McCulloch and Pitts [15]. The actual adaptable perceptron model is credited to

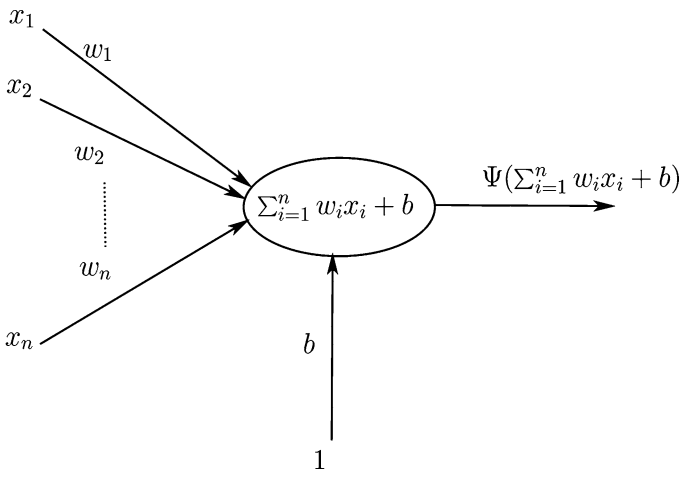

Fig. 2 Perceptron

Rosenblatt [25]. The perceptron is a simple single layer neuron model, which uses a learning rule similar to gradient descent. However, the simplicity of this model (single layer) limits its applicability to model complex practical problems; thereby, it was an object of censure in [19]. However, a question which instigated the use of multilayer neural networks was kindled in [19]. After a couple of decades of research, neural network research exploded with impressive success. Furthermore, multilayered feedforward neural networks are rigorously established as a function class of universal approximators [11]. In addition to that, different models of ANNs were proposed to solve combinatorial optimization problems. Furthermore, the convergence conditions for the ANNs optimization models have been extensively analyzed [31].

Processing elements (PEs) are the primary elements of any ANN. The state of PE can take any real value between the interval $[0, 1]$ (some authors prefer to use the values between $[-1,1]$; however, both definitions are interchangeable and have the same convergence behavior). The main characteristic of a PE is to do function embedding. In order to understand this phenomenon, consider a single PE ANN model (the perceptron model) with n inputs and one output, shown in Fig. 2.

The total information incident on the PE is $\sum_{i=1}^{n} w_i x_i$. PE embeds this information into a transfer function Ψ, and sends the output to the following layer. Since there is a single layer in the example, the output from the PE is considered as the final output. Moreover, if we define Ψ as

$$\Psi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i + b \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where b is the threshold level of the PE, then the single PE perceptron can be used for binary classification, given the data is linearly separable. The difference

between this simple perceptron method of classification, and support vector-based classification is that the perceptron finds a plane that linearly separates the data; however, support vector finds the plane with maximum margin. This does not indicate superiority of one method over the other method since a single PE is considered. In fact, this shows the capability of a single PE; however, a single PE is incapable to process complex information that is required for most practical problems. Therefore, multiple PEs in multiple layers are used as universal classifiers. The PEs interact with each other via links to share the available information. The intensity and sense of interactions between any two connecting PEs is represented by weight (or synaptic weight, the term synaptic is related to the nervous system, and is used in ANNs to indicate the weight between any two PEs) on the links.

Usually, PEs in the $(r-1)^{\text{th}}$ layer send information to the r^{th} layer using the following feedforward rule:

$$y_i = \Psi_i \left(\sum_{j \in (r-1)} w_{ji} y_j - U_i \right), \tag{19}$$

where PE i belongs to the r^{th} layer, and any PE j belongs to the $(r-1)^{\text{th}}$ layer. y_i represents the state of the i^{th} PE, w_{ji} represents weight between the j^{th} PE and i^{th} PE, and U_i represents threshold level of the i^{th} PE. Function $\Psi_i()$ is the transfer function for the i^{th} PE. Once the PEs in the final layer are updated, the error from the actual output is calculated using a loss function (this is the part where correntropic loss function will be injected). The error or loss calculation marks the end of feed forward phase of ANNs. Based on the error information, back-propagation phase of ANNs starts. In this phase, the error information is utilized to update the weights, using the following rules:

$$w_{jk} = w_{jk} + \mu \, \delta_k \, y_j, \tag{20}$$

where

$$\delta_k = \frac{\partial \mathcal{F}(\varepsilon)}{\partial \varepsilon_n} \, \Psi'(net_k), \tag{21}$$

where μ is the learning step size, $net_k = \sum_{j \in (r-1)} w_{ji} y_j - U_k$, and $\mathcal{F}(\varepsilon)$ is the error function (or loss function). For the output layer, the weights are computed as

$$\begin{aligned} \delta_k = \delta_0 &= \frac{\partial \mathcal{F}(\varepsilon)}{\partial \varepsilon} \, \Psi'(net_k), \\ &= (y - y_0) \, \Psi'(net_k), \end{aligned} \tag{22}$$

and the deltas of the previous layers are updated as

$$\delta_k = \delta_h = \Psi'(net_k) \sum_{o=1}^{N_0} w_{ho} \delta_o. \tag{23}$$

In the proposed approaches, ANN is trained in order to minimize the correntropic loss function. In total, two different approaches to train ANN are proposed. In one approach, ANN will be trained using the CS algorithm. Whereas in the other proposed approach, ANN will be trained using the SA algorithm. In order to validate our results, we will not only compare the proposed approaches with conventional ANN training methods, but also compare them with the support vector machines based classification method. In the following section, a review of support vector machines is presented.

2.6 Support Vector Machines

A support vector machine (SVM) is a popular supervised learning method [5, 7]. It has been developed for binary classification problems, but can be extended to multiclass classification problems [9,33,36] and it has been applied in many areas of engineering and biomedicine [10,12,21,32,37]. In general supervised classification algorithms provide a classification rule able to decide the class of an unknown sample. In particular the goal of SVMs training phase is to find a hyperplane that "optimally" separates the data samples that belong to a class. More precisely SVM is a particular case of hyperplane separation. The basic idea of SVM is to separate two classes (say A and B) by a hyperplane defined as

$$f(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + b, \tag{24}$$

such that $f(\mathbf{a}) < 0$ when $\mathbf{a} \in A$, and $f(\mathbf{b}) > 0$ when $\mathbf{b} \in B$. However, there could be infinitely many possible ways to select $\mathbf{w}$. The goal of SVM is to choose a best $\mathbf{w}$ according to a criterion (usually the one that maximizes the margin), so that the risk of misclassifying a new unlabeled data point is minimum. A best separating hyperplane for unknown data will be the one, that is sufficiently far from both the classes (it is the basic notion of SRM), i.e., a hyperplane which is in the middle of the following two parallel hyperplanes (support hyperplanes) can be used as a separating hyperplane:

$$\mathbf{w}^t\mathbf{x} + b = c, \tag{25}$$

$$\mathbf{w}^t\mathbf{x} + b = -c. \tag{26}$$

Since, $\mathbf{w}$, b, and c are all parameters, a suitable normalization will lead to

$$\mathbf{w}^t\mathbf{x} + b = 1, \tag{27}$$

$$\mathbf{w}^t\mathbf{x} + b = -1. \tag{28}$$

Moreover, the distance between the supporting hyperplanes (27) and (28) is given by

$$\Delta = \frac{2}{||\mathbf{w}||}. \tag{29}$$

In order to obtain the best separating hyperplane, the following optimization problem is solved:

maximize:

$$\frac{2}{||\mathbf{w}||} \tag{30a}$$

subject to:

$$y_i(\mathbf{w}^t\mathbf{x_i} + b) - 1 \geq 0 \quad \forall i. \tag{30b}$$

The objective given in (30a) is replaced by minimizing $||\mathbf{w}||^2/2$. Usually, the solution to problem (30) is obtained by solving its dual. In order to obtain the dual, consider the Lagrangian of (30), given as

$$\mathcal{L}(\mathbf{w}, b, \mathbf{u}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} u_i \left(y_i(\mathbf{w}^t\mathbf{x_i} + b) - 1\right), \tag{31}$$

where $u_i \geq 0 \; \forall \; i$. Now, observe that problem (30) is convex. Therefore, the strong duality holds, and the following equation is valid:

$$\min_{(\mathbf{w},b)} \max_{\mathbf{u}} \mathcal{L}(\mathbf{w}, b, \mathbf{u}) = \max_{\mathbf{u}} \min_{(\mathbf{w},b)} \mathcal{L}(\mathbf{w}, b, \mathbf{u}). \tag{32}$$

Moreover, from the saddle point theory [4], the following hold:

$$\mathbf{w} = \sum_{i=1}^{N} u_i y_i \mathbf{x_i}, \tag{33}$$

$$\sum_{i=1}^{N} u_i y_i = 0. \tag{34}$$

Therefore, using (33) and (34), the dual of (30) is given as

maximize:

$$\sum_{i=1}^{N} u_i - \frac{1}{2} \sum_{i,j=1}^{N} u_i u_j y_i y_j \mathbf{x_i}^t \mathbf{x_j} \tag{35a}$$

subject to:

$$\sum_{i=1}^{N} u_i y_i = 0, \tag{35b}$$

$$u_i \geq 0 \quad \forall i. \tag{35c}$$

Thus, solving (35) results in obtaining support vectors, which in turn leads to the optimal hyperplane. This phase of SVM is called as training phase. The testing phase is simple and can be stated as

$$y_{\text{test}} = \begin{cases} -1, \text{ test } \in A & \text{if } f^*(\mathbf{x_{test}}) < 0, \\ +1, \text{ test } \in B & \text{if } f^*(\mathbf{x_{test}}) > 0. \end{cases} \tag{36}$$

The above method works very well when the data is linearly separable. However, most of the practical problems are not linearly separable. In order to extend the usability of SVMs, soft margins and kernel transformation are incorporated in the basic linear formulation. When considering soft margin, (30a) is modified as

$$y_i(\mathbf{w}^t\mathbf{x_i} + b) - 1 + s_i \geq 0 \quad \forall i, \tag{37}$$

where $s_i \geq 0$ are slack variables. The primal formulation is then updated as

minimize:

$$\frac{1}{2}||\mathbf{w}||^2 + c\sum_{i=1}^{N} s_i \tag{38a}$$

subject to:

$$y_i(\mathbf{w}^t\mathbf{x_i} + b) - 1 + s_i \geq 0 \quad \forall i, \tag{38b}$$

$$s_i \geq 0 \quad \forall i. \tag{38c}$$

Similar to the linear SVM, the Lagrangian of formulation (38) is given by

$$\mathcal{L}(\mathbf{w}, b, \mathbf{u}, \mathbf{v}) = \frac{1}{2}||\mathbf{w}||^2 + c\sum_{i=1}^{N} s_i - \sum_{i=1}^{N} u_i \left(y_i(\mathbf{w}^t\mathbf{x_i} + b) - 1\right) - \mathbf{v}^t\mathbf{s}, \tag{39}$$

where $u_i, v_i \geq 0 \; \forall \; i$. Correspondingly, using the theory of saddle point and strong duality, the soft margin SVM dual is defined as

maximize:

$$\sum_{i=1}^{N} u_i - \frac{1}{2}\sum_{i,j=1}^{N} u_i u_j y_i y_j \mathbf{x_i}^t\mathbf{x_j} \tag{40a}$$

subject to:

$$\sum_{i=1}^{N} u_i y_i = 0, \tag{40b}$$

$$u_i \leq c \quad \forall i, \tag{40c}$$

$$u_i \geq 0 \quad \forall i. \tag{40d}$$

Furthermore, the dot product "$\mathbf{x_i}^t\mathbf{x_j}$" in (40a) is exploited to overcome the nonlinearity, i.e., by using kernel transformations into a higher dimensional space. Thus, the soft margin kernel SVM has the following dual formulation:

maximize:

$$\sum_{i=1}^{N} u_i - \frac{1}{2}\sum_{i,j=1}^{N} u_i u_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \tag{41a}$$

subject to:

$$\sum_{i=1}^{N} u_i y_i = 0, \tag{41b}$$

$$u_i \leq c \quad \forall i, \tag{41c}$$

$$u_i \geq 0 \quad \forall i, \tag{41d}$$

where $K(.,.)$ is any symmetric kernel. In this chapter, a Gaussian kernel is used, which is defined as

$$K(\mathbf{x_i}, \mathbf{x_j}) = \mathrm{e}^{-\gamma||\mathbf{x_i}-\mathbf{x_j}||^2}, \tag{42}$$

where $\gamma > 0$. Therefore, in order to classify the data, two parameters (c, γ) should be given a priori. The information about the parameters is obtained from the knowledge and structure of the input data. However, this information is intangible for practical problems. Thus, an exhaustive logarithmic grid search is conducted over the parameter space to find their suitable values. It is worthwhile to mention that assuming c and γ as variables for the kernel SVM, and letting the kernel SVM try to obtain the optimal values of c and γ, makes the classification problem (41) intractable.

Once the parameter values are obtained from the grid search, the kernel SVM is trained to obtain the support vectors. Usually the training phase of the kernel SVM is performed in combination with a re-sampling method called cross validation. During cross validation the existing data set is partitioned in two parts (training and testing). The model is built based on the training data, and its performance is evaluated using the testing data. In [28], a general method to select data for training SVM is discussed. Different combinations of training and testing sets are used to calculate average accuracy. This process is mainly followed in order to avoid manipulation of classification accuracy results due to a particular choice of the training and testing datasets. Finally the classification accuracy reported is the average classification accuracy for all the cross validation iterations. There are several cross validation methods available to built the training and testing sets. Next, three most common methods of cross validation are described:

- k-Fold cross validation (kCV): In this method, the data set is partitioned in k equally sized groups of samples (folds). In every cross validation iteration $k-1$ folds are used for the training and 1 fold is used for the testing. In the literature usually k takes a value from $1, \ldots, 10$.

- Leave one out cross validation (LOOCV): In this method, each sample represents one fold. Particularly, this method is used when the number of samples are small, or when the goal of classification is to detect outliers (samples with particular properties that do not resemble the other samples of their class).
- Repeated random subsampling cross validation (RRSCV): In this method, the data set is partitioned into two random sets, namely training set and validation (or testing) set. In every cross validation, the training set is used to train the SVM and the testing (or validation) set to test the accuracy of SVM. This method is preferred, if there are large number of samples in the data. The advantage of this method (over k-fold cross validation) is that the proportion of the training set and number of iterations are independent. However, the main drawback of this method is, if few cross validations are performed, then some observations may never be selected in the training phase (or the testing phase), whereas others may be selected more than once in the training phase (or the testing phase, respectively). To overcome this difficulty, the kernel SVM is cross validated sufficiently large number of times, so that each sample is selected atleast once for training as well as testing the kernel SVM. These multiple cross validations also exhibits Monte Carlo variation (since the training and testing sets are chosen randomly).

In this chapter, the RRSCV method is used to train the kernel SVM, the performance accuracy of the SVM is compared with the proposed approaches. In the next section, the different learning methodologies used to train ANNs and SVMs will be discussed.

3 Obtaining an Optimal Classification Rule Function

The goal of any learning algorithm is to obtain the optimal rule $f^\star$ by solving the classification problem illustrated in formulation (6). Based on the type of loss function used in risk estimation, the type of information representation, and the type of optimization algorithm, different classification algorithms can be designed. In this section, five different classification methods, two of them are novel and the rest are conventional methods (used for comparison with novel methods), will be discussed. A summary of the classification methods is listed in Table 1. In the following part of this section, each of the listed methods will be explained.

3.1 Conventional Nonparametric Approaches

A classical method of classification using ANN involves training a multilayer perceptron (MLP) using a back-propagation algorithm. Usually, a signmodal function is used as an activation function, and a quadratic loss function is used for error

Table 1 Notation and description of proposed (✠) and existing (✓) methods

Notation	Information representation	Loss function	Optimization algorithm
AQG✓	Nonparametric (ANN)	Quadratic	Exact method— gradient descent
ACG ✓	Nonparametric (ANN)	Initially quadratic, shifts to correntropy with fixed kernel width	Exact method— gradient descent
ACC✠	Nonparametric (ANN)	Correntropy with varying kernel width	Heuristic method— convolution smoothing
ACS✠	Nonparametric (ANN)	Correntropy with fixed kernel width	Heuristic method— simulated annealing
SGQ✓	Parametric (SVM)	Quadratic with Gaussian kernel	Exact method— quadratic optimization

measurement. The ANN is trained using a back-propagation algorithm involving gradient descent method [16]. Before proceeding further to present the training algorithms, let us define the notations:

w_{jk}^n: The weight between the k^{th} and j^{th} PEs at the n^{th} iteration.
y_j^n: Output of the j^{th} PE at the n^{th} iteration.
$net_k^n = \sum_j w_{jk}^n y_j^n$: Weighted sum of all outputs y_j^n of the previous layer at n^{th} iteration.
$\Psi()$: Sigmoidal squashing function in each PE, defined as:

$$\Psi(z) = \frac{1 - \mathrm{e}^{-2}}{1 + \mathrm{e}^{-2}}.$$

$y_k^n = \Psi(net_k^n)$: Output of k^{th} PE of the current layer, at the n^{th} iteration.
$y^n \in \{\pm 1\}$: The true label (actual label) for the n^{th} sample.

In the following part of this section, two training algorithms (AQG and ACG) will be described. These algorithms differ in the type of loss function used to train ANNs.

3.1.1 Training ANN with Quadratic Loss Function Using Gradient Descent

Training ANN with quadratic loss function using gradient descent (AQG) is the simplest and most widely known method of training ANN. A three layered ANN (input, hidden, and output layers) is trained using a back-propagation algorithm. Specifically, the generalized delta rule is used to update the weights of ANN, and the training equations are

$$w_{jk}^{n+1} = w_{jk}^n + \mu\, \delta_k^n\, y_j^n, \tag{43}$$

where

$$\delta_k^n = \frac{\partial MSE(\varepsilon)}{\partial \varepsilon_n} \Psi'(net_k^n), \tag{44}$$

where μ is the learning step size, $\varepsilon = (y^n - y_0^n)$ is the error (or loss), and $MSE(\varepsilon)$ is the mean square error. For the output layer, the weights are computed as

$$\begin{aligned} \delta_k^n = \delta_0^n &= \frac{\partial MSE(\varepsilon)}{\partial \varepsilon_n} \Psi'(net_k^n), \\ &= (y^n - y_0^n)\, \Psi'(net_k^n), \end{aligned} \tag{45}$$

The deltas of the previous layers are updated as

$$\delta_k^n = \delta_h^n = \Psi'(net_k^n) \sum_{o=1}^{N_0} w_{ho}^n \delta_o^n. \tag{46}$$

3.1.2 Training ANN with Correntropic Loss Function Using Gradient Descent

Training ANN with correntropic loss function using gradient descent (ACG) method is similar to AQG method, the only difference is the use of correntropic loss function instead of quadratic loss function. Furthermore, the kernel width of correntropic loss is fixed to a smaller value (in [29], a value of 0.5 is illustrated to perform well). Moreover, since the correntropic function is non-convex at that kernel width, the ANN is trained with a quadratic loss function for some initial epochs. After sufficient number of epochs (ACG_1), the loss function is changed to correntropic loss function. Thus (ACG_1) is a parameter of the algorithm. The reason for using quadratic loss function at the initial epochs is to prevent converging at a local minimum at early learning stages. Similar to AQG, the delta rule is used to update the weights of ANN, and the training equations are:

$$w_{jk}^{n+1} = w_{jk}^n + \mu\, \delta_k^n\, y_j^n, \tag{47}$$

where

$$\delta_k^n = \frac{\partial \mathcal{F}(\varepsilon)}{\partial \varepsilon_n} \Psi'(net_k^n), \tag{48}$$

where μ is the step length, and $\mathcal{F}(\varepsilon)$ is a general loss function, which can be either quadratic or correntropic function based on the current number of training epochs. For the output layer, the weights are computed as

$$\delta_k^n = \delta_0^n = \frac{\partial \mathcal{F}(\varepsilon)}{\partial \varepsilon_n} \Psi'(net_k^n)$$
$$= \begin{cases} \frac{\beta}{\sigma^2} \mathrm{e}\left(\frac{-(y^n - y_0^n)^2}{2\sigma^2}\right) (y^n - y_0^n)\, g'(net_k^n) & \text{if } \mathcal{F} \equiv \text{C-loss function}, \\ (y^n - y_0^n)\, g'(net_k^n) & \text{if } \mathcal{F} \equiv \text{MSE function}, \end{cases} \tag{49}$$

where C-loss is the correntropic loss. The deltas of the previous layers are updated as

$$\delta_k^n = \delta_h^n = \Psi'(net_k^n) \sum_{o=1}^{N_0} w_{ho}^n \delta_o^n. \tag{50}$$

Based on the results of [29], the value of ACG_1 is taken as five epochs. The purpose of comparing the proposed approaches with the ACG method is to see the improvement in the classification accuracy, when the kernel width is changing smoothly.

3.2 Conventional Parametric Approach

3.2.1 Training Soft Margin SVM with Gaussian Kernel (SGK)

SVM is one of the most widely known parametric methods in classification. A general framework of SVM is presented in the earlier section. In the present work, a Gaussian kernel-based soft margin SVM is used. The SVM is implemented in two steps. In the first step, optimal parameters (kernel width and cost penalty) are obtained via exhaustive searching over the parameter space. Once the optimal parameters are obtained, in the second step, the kernel SVM is trained with the optimal parameters.

From the grid search, appropriate values of the parameters are selected. Based on the selected values of the parameters, the SVM is trained with 100 Monte Carlo simulations. In each simulation, a data is divided into two random subsets for training and testing (RRSCV method). The use of the kernel SVM in this chapter is to compare the results of our proposed algorithms. Next, the proposed algorithms are presented.

3.3 Proposed Nonparametric Approaches

In this section, two optimization methods that utilize the correntropic loss function are proposed. In one of the methods, the kernel width act as variable. Whereas, in the other method, the kernel width is set as a parameter. In the following part of the section, the two proposed methods will be discussed.

3.3.1 Training ANN with Correntropic Loss Function Using Convolution Smoothing (ACC)

Similar to the previous ANN-based methods, a back-propagation algorithm is used to train the ANN, i.e., in this method, the weights are updated using the delta rule. However, the cost function $\mathcal{F}$ is always the correntropic function, and the kernel width σ is changed over the training period. The kernel width act as a smoothing parameter of the CS algorithm, and initially kernel width is set to a value of 2. As the algorithm proceeds, the kernel width is smoothly reduced till it reaches 0.5. Furthermore, as the algorithm progress, if the delta rule leads to a high error value, then the kernel width is increased to a value of 2 with probability P_{accept}, to escape from the local minima. This probability is reduced exponentially depending on the number of epochs. ACC method can be seen as a stochastic CS method which minimizes the correntropic loss function. The training equations for the underlying ANN framework are as follows:

$$w_{jk}^{n+1} = w_{jk}^{n} + \mu\, \delta_k^n y_j^n, \tag{51}$$

where for the output layer, the deltas and weights are computed as

$$\delta_k^n = \frac{\partial \mathcal{F}_C^{\sigma}(\varepsilon)}{\partial \varepsilon_n}\, \Psi'(net_k^n), \tag{52}$$

$$\delta_k^n = \delta_0^n = \frac{\partial \mathcal{F}_C^{\sigma}(\varepsilon)}{\partial \varepsilon_n}\, \Psi'(net_k^n), \tag{53}$$

$$= \frac{\beta}{\sigma^2}\, \mathrm{e}\left(\frac{-(y^n - y_0^n)^2}{2\sigma^2}\right) (y^n - y_0^n)\, \Psi'(net_k^n), \tag{54}$$

where $\mathcal{F}_C^{\sigma} \equiv$ correntropic loss function with kernel width σ, and $\mathcal{F}_C^{\sigma}(\varepsilon)$ is the error at the output layer. The deltas of the previous layers are updated as

$$\delta_k^n = \delta_h^n = \Psi'(net_k^n) \sum_{o=1}^{N_0} w_{ho}^n \delta_o^n. \tag{55}$$

The ACC method is illustrated in Algorithm 1 for a given $n \times p$ data matrix with r elements in the middle layer. Algorithm 1 represents ACC learning method for the block update scenario. For the sample by sample update scenario, Algorithm 1 is adjusted appropriately to incorporate the CS mechanism.

In Algorithm 1, σ_0, α_1 are the parameters that control the flow of ACC method, and their values are taken as $2,\ 0.5\mathbf{e}$, respectively (where $\mathbf{e}$ is vector of ones). $f_1,\ f_2$ are the functions to update σ, P_{accept} is the probability of accepting noisy solutions. For the sake of simplicity, f_1 and P_{accept} are taken as exponentially decreasing functions, and f_2 updates σ to a value of 2.

Algorithm 1 ACC method

```
Setup the ANN (structure and transfer functions)
Randomly initialize w_jk^0 ∈ {0, 1}
Set σ = σ_0, μ = μ_0
while termination criteria do
  BLOCK FEEDFORWARD PHASE—ANN
  if F_C^σ(") < F_C^σ(α_1) then
    σ = f_1(σ)
  else
    if random() < P_accept then
      σ = f_1(σ)
    else
      σ = f_2(σ)
    end if
  end if
  if F_C^σ(") < minErr then
    termination criteria = true
  end if
  BLOCK BACKPROPAGATION PHASE - ANN
end while
```

3.3.2 Training ANN with Correntropic Loss Function Using Simulated Annealing (ACS)

Unlike the previous gradient descent based learning methods, in this method a SA algorithm is used to train ANN, i.e., no gradient search is involved in ANN. This method assumes that the correntropic loss function has a fixed kernel width. Since the kernel width determines the convexity of the loss function, a gradient descent method cannot be used as a learning method in a generalized framework. Hence, the SA algorithm is used as a learning method to avoid convergence to a local minimum. The ACS method is illustrated in Algorithm 2 for a given $n \times p$ data matrix with r elements in the middle layer. Furthermore, $\sigma = \widehat{\sigma}$ is a given parameter of the algorithm. Moreover, the ACS algorithm is used in block update mode only, unlike the ACC algorithm (i.e., ACC algorithm can be used in a sample or block-based update mode).

In Algorithm 2, T_0 is the initial temperature, and its value is taken as 1. $f_1(T)$ and $P_{\text{accept}}(T)$ are two different functions of temperature. $f_1(T)$ is a simple exponential cooling function, whereas function $P_{\text{accept}}(T)$ is exponential probability, which depends upon the values of T, $\Delta_{\mathfrak{O}}$, and $\Delta_{\mathfrak{O}-1}$. There are two termination criteria for ACC and ACS method. Either the total error should fall below *minErr* (taken as 0.001) or the number of epochs should exceed *MaxEpochs* (*MaxEpochs* is a parameter for experimental runs, and is varied from $1, \ldots, 10$).

Algorithm 2 ACS method

Setup the ANN (structure and transfer functions)
Randomly initialize $W_0 = \{w_{jk}^0 \in \{0, 1\}\} \forall\ i, j$ connections in ANN.
Initialize $\eth = 0$ and $T = T_0$
$\Delta_0 = \mathcal{F}_C^{\sigma}(\boldsymbol{\varepsilon}_0)$
while termination criteria **do**
  $T = f_1(T)$
  $\eth = \eth + 1$
  $W_{\eth} = neighbor(W_{\eth-1})$
  BLOCK FEEDFORWARD PHASE—ANN
  $\Delta_{\eth} = \mathcal{F}_C^{\sigma}(\boldsymbol{\varepsilon}_{\eth})$
  if $\Delta_{\eth} < minErr$ **then**
    $termination\ criteria = true$
  end if
  if $\Delta_{\eth} < \Delta_{\eth-1}$ **then**
    $W_{\eth} = W_{\eth-1}$
    $\Delta_{\eth} = \Delta_{\eth-1}$
  else
    if $random() < P_{accept}(T)$ **then**
      $W_{\eth} = W_{\eth-1}$
      $\Delta_{\eth} = \Delta_{\eth-1}$
    end if
  end if
end while

4 Experimental Results and Discussion

Simulations are carried out for three real-world data sets (Wisconsin Breast Cancer Data, Pima Indians Diabetes Data, and BUPA Liver Disorder Data) related to biomedical field. These data sets are taken from the UCI machine-learning repository.[3] A brief information regarding each of the data sets is given in Table 2.

Originally, some of the selected data sets have missing values. All the records containing any missing data values are discarded before using the data for classification. In addition to that, for each data set, a fixed number of records were selected for training the classifier. The remaining records were used for testing the trained classifier. In order to have accurate results, a data set is randomly divided into testing data and training data (keeping the total number of training records fixed, as given in Table 2). For each data set, the training data is preprocessed by normalizing the data to zero mean and unit variance along the features (to avoid scaling effect). Based on the mean and variance of the training data, the testing data is scaled. The purpose of normalizing the training data alone, and scaling the testing data later, is to mimic the real life scenario. Usually, the testing data is not available beforehand, and its information is unknown while normalizing the training data. In addition to

[3] http://archive.ics.uci.edu/ml/.

Table 2 Data sets used in the experimental study

Data set	Attributes (or) features	Total records	Classes	Training size
Pima Indians Diabetes (PID)	8	768	2	400
Wisconsin Breast Cancer (WBC)	9	683	2	300
BUPA Liver Disorders (BLD)	6	345	2	150

that, for the results to be as consistent as possible, 100 Monte Carlo simulations were conducted (both for ANN and SVM), and the average testing accuracy of the classifier over the 100 simulations is reported in the results.

Comparison Among ANN-Based Methods

Since the number of PEs in the hidden layer has an effect on the performance of ANN-based classifiers, experiments have been conducted for 5, 10, and 20 PEs in the hidden layer for each of the data sets. Although the exact number of PEs that will give maximum classification accuracy is unknown, it can be estimated by an experimental search over the number of PEs in the hidden layer. However, such a search is out of the scope of the current work due to its high computational requirements. Therefore, we have confined our computations for 5, 10, and 20 PEs in order to efficiently compare all the ANN-based classifiers. Moreover, the performance of ANN-based classifier with sample and block-based learning framework was also considered in the comparison. This is mainly done to find whether the proposed algorithms perform better in sample or block-based learning mode. The results of sample and block-based learning methods of ANN simulations are given in Tables 3, 4, and 5.

In order to present the results with different parameters, a similar pattern is followed in Tables 3, 4, and 5. Each column represents a number of learning *epochs* for sample-based learning. Whereas, each column represents a number of *epochs*$\times$*training sample size* for block-based learning. For a given algorithm, a row represents the average result of 100 Monte Carlo simulations. First row presents the results with 5 PEs in hidden layer. Second row presents the results with 10 PEs, and third row presents the results with 20 PEs in the hidden layer.

For the AQG and ACG methods, the results from [29] are used as a reference for further comparisons (see Tables 3a, 4a, and 5a). Since ACS requires knowledge of change in loss function value over any two consecutive iterations, it cannot be implemented in sample-based learning. However, all the algorithms have been implemented in block-based learning. Moreover, in all the tables, the performance results of ACS at $\sigma = 0.5$ have been presented. The results show that ACC almost always (both for sample and block-based learning methods) performs better when compared to any of the ANN-based classification algorithm. Thus, this method can be used as a generalized robust ANN-based classifier for practical data classification problems. Moreover, the poor performance of ACS method is attributed to the

Table 3 Performance of ANN on PID Data

(a) Sample based learning

	1	2	3	4	5	6	7	8	9	10	Best
AQG Accuracy	0.7399	0.7549	0.7572	0.7571	0.7568	0.7564	0.7563	0.7548	0.754	0.7535	
	0.7496	0.7571	0.7576	0.7574	0.7565	0.7563	0.7559	0.7553	0.7551	0.7543	0.7576
	0.7383	0.7442	0.7441	0.7482	0.7478	0.7465	0.7451	0.7431	0.7442	0.7437	
ACG Accuracy	–	–	–	–	–	0.7623	0.7627	0.7625	0.7628	0.7631	
	–	–	–	–	–	0.7646	0.7661	0.7658	0.765	0.7645	0.7661
	–	–	–	–	–	0.759	0.7609	0.7604	0.7599	0.7597	
ACC Accuracy	0.6865	0.746	0.7535	0.7632	0.7612	0.7609	0.7626	0.7621	0.7616	0.7607	
	0.7308	0.7584	0.7603	0.7654	0.7639	0.7646	0.7679	0.766	0.7647	0.7637	**0.7679**
	0.7474	0.7587	0.764	0.7623	0.7643	0.763	0.7599	0.7653	0.7657	0.7625	

(b) Block based learning

	1	2	3	4	5	6	7	8	9	10	Best
AQG Accuracy	0.7099	0.7441	0.758	0.7628	0.7637	0.7624	0.7671	0.7639	0.763	0.7662	
	0.7358	0.7562	0.7622	0.763	0.7661	0.7666	0.7685	0.7665	0.766	0.765	0.7685
	0.7643	0.7613	0.7658	0.7684	0.7645	0.7676	0.7669	0.7665	0.7622	0.7641	
ACG Accuracy	–	–	–	–	–	0.7659	0.7666	0.7645	0.7647	0.7655	
	–	–	–	–	–	0.7666	0.7651	0.7694	0.7652	0.7652	0.7694
	–	–	–	–	–	0.7672	0.7652	0.7652	0.7653	0.7656	
Acc Accuracy	0.6703	0.7007	0.724	0.7405	0.7524	0.7587	0.7587	0.7651	0.7652	0.7622	
	0.6976	0.7253	0.7456	0.7542	0.7618	0.763	0.7645	0.7667	0.7698	0.7676	**0.7698**
	0.7196	0.7502	0.7624	0.7625	0.7641	0.767	0.7643	0.7646	0.766	0.7693	
ACS Accuracy	0.7548	0.7524	0.7519	0.7559	0.7543	0.7511	0.7544	0.7535	0.753	0.7495	
	0.7523	0.7517	0.7546	0.7489	0.7531	0.7516	0.7498	0.7505	0.7497	0.7549	0.7556
	0.7474	0.7543	0.7514	0.7523	0.7484	0.7502	0.7485	0.7477	0.748	0.7501	

Table 4 Performance of ANN on BLD Data

(a) Sample based learning

	1	2	3	4	5	6	7	8	9	10	Best
AQG Accuracy	0.5692	0.5777	0.5782	0.5771	0.5875	0.5839	0.5911	0.5897	0.5912	0.5918	
	0.568	0.5697	0.5842	0.5835	0.5955	0.5986	0.5985	0.6063	0.6107	0.6104	0.6196
	0.5723	0.5737	0.5908	0.5972	0.603	0.6112	0.6028	0.6144	0.6196	0.6222	
ACG Accuracy	–	–	–	–	–	0.5777	0.579	0.5789	0.5803	0.5828	
	–	–	–	–	–	0.5787	0.5805	0.5849	0.5846	0.5873	0.596
	–	–	–	–	–	0.5838	0.596	0.5935	0.5945	0.5922	
ACC Accuracy	0.575	0.5766	0.5809	0.5791	0.5831	0.5848	0.5924	0.5895	0.5919	0.5968	
	0.5698	0.5759	0.5816	0.584	0.5913	0.5913	0.5974	0.6025	0.6101	0.613	**0.6271**
	0.5706	0.5811	0.5822	0.5918	0.6013	0.5999	0.6083	0.6121	0.6223	0.6271	

(b) Block based learning

	1	2	3	4	5	6	7	8	9	10	Best
AQG Accuracy	0.5614	0.5698	0.5899	0.5967	0.5945	0.6023	0.613	0.6152	0.6307	0.6381	
	0.5697	0.5951	0.5961	0.6099	0.6248	0.6379	0.6436	0.653	0.6474	0.6577	0.6845
	0.5797	0.6097	0.6374	0.6437	0.6517	0.6628	0.6724	0.6681	0.6746	0.6845	
ACG Accuracy	–	–	–	–	–	0.6121	0.6138	0.6154	0.6258	0.6326	
	–	–	–	–	–	0.6306	0.6391	0.6427	0.6549	0.6592	0.6854
	–	–	–	–	–	0.6595	0.6669	0.6714	0.6746	0.6854	
ACC Accuracy	0.5703	0.5777	0.5814	0.591	0.6037	0.6044	0.6281	0.6303	0.6316	0.6409	
	0.5649	0.5848	0.5976	0.6165	0.6215	0.6308	0.6503	0.6471	0.6587	0.6682	**0.6862**
	0.5812	0.6079	0.6335	0.6386	0.6619	0.663	0.6673	0.6753	0.6772	0.6862	
ACS Accuracy	0.6124	0.6337	0.6426	0.637	0.636	0.6411	0.6331	0.646	0.6426	0.6423	
	0.6367	0.6548	0.655	0.6568	0.6559	0.6596	0.6576	0.6574	0.6556	0.6542	0.6753
	0.6526	0.6753	0.6676	0.6689	0.6683	0.6688	0.6642	0.6738	0.6697	0.6695	

Table 5 Performance of ANN on WBC Data

(a) Sample based learning

	1	2	3	4	5	6	7	8	9	10	Best
AQG Accuracy	0.9656	0.9682	0.9688	0.9689	0.9689	0.9688	0.9689	0.9684	0.968	0.9678	
	0.9646	0.9693	0.9695	0.9695	0.9694	0.9695	0.9694	0.9691	0.9689	0.9685	0.9699
	0.9645	0.9695	0.9699	0.9696	0.9699	0.9693	0.969	0.9687	0.9683	0.968	
ACG Accuracy	–	–	–	–	–	0.9695	0.9696	0.9696	0.9699	0.9699	
	–	–	–	–	–	0.9698	0.9699	0.9704	0.9704	0.9705	0.9707
	–	–	–	–	–	0.9702	0.9705	0.9707	0.9707	0.9707	
ACC Accuracy	0.9693	0.9698	0.9693	0.969	0.9692	0.97	0.9706	0.9702	0.9703	0.9702	
	0.9699	0.9699	0.9713	0.9696	0.9715	0.9702	0.9696	0.9707	0.971	0.9713	**0.9715**
	0.971	0.9701	0.9703	0.9701	0.9696	0.9696	0.9704	0.9701	0.9691	0.9696	

(b) Block based learning

	1	2	3	4	5	6	7	8	9	10	Best
AQG Accuracy	0.9605	0.9682	0.968	0.9677	0.97	0.97	0.9696	0.9702	0.969	0.9695	
	0.9643	0.9677	0.9682	0.9684	0.9687	0.9704	0.9691	0.9698	0.969	0.9683	0.9704
	0.9658	0.9695	0.9694	0.969	0.9692	0.9689	0.97	0.9696	0.9701	0.9695	
ACG Accuracy	–	–	–	–	–	0.9697	0.9666	0.9689	0.9666	0.9695	
	–	–	–	–	–	0.9705	0.9702	0.971	0.9726	0.965	0.9726
	–	–	–	–	–	0.9705	0.9653	0.97	0.9689	0.9705	
ACC Accuracy	0.9614	0.9663	0.968	0.9684	0.9695	0.9706	0.9701	0.9697	0.9715	0.9685	
	0.9645	0.9678	0.9686	0.9703	0.9683	0.969	0.9684	0.9696	0.9698	0.9688	**0.9715**
	0.9656	0.9681	0.969	0.9705	0.969	0.9697	0.9697	0.9698	0.9696	0.9695	
ACS Accuracy	0.9652	0.9651	0.9659	0.965	0.9644	0.9647	0.9643	0.9647	0.9672	0.9656	
	0.9644	0.9659	0.9665	0.9645	0.9651	0.9652	0.9643	0.9647	0.965	0.9657	0.9672
	0.9648	0.9639	0.9649	0.9639	0.9639	0.9622	0.9631	0.9634	0.9654	0.9638	

Table 6 Performance of ACS for different values of σ and the number of PEs in hidden layer

PID Data						
	0.5	0.8	1	1.2	1.4	1.6
5 PE	0.7556	0.749	0.7593	0.7568	0.7633	0.7616
10 PE	0.7549	0.7461	0.7585	0.7604	0.7608	0.7603
20 PE	0.7543	0.7423	0.7614	0.758	0.7585	0.7593
BLD Data						
	0.5	0.8	1	1.2	1.4	1.6
5 PE	0.646	0.6806	0.681	0.6861	0.6853	0.684
10PE	0.6596	0.6884	0.6928	0.6931	0.6941	0.6928
20 PE	0.6753	0.6992	0.6996	0.6997	0.7013	0.7007
WBC Data						
	0.5	0.8	1	1.2	1.4	1.6
5 PE	0.9672	0.9646	0.9633	0.9648	0.9648	0.9672
10 PE	0.9665	0.9639	0.9631	0.9647	0.9634	0.9635
20 PE	0.9654	0.9621	0.9613	0.9625	0.963	0.9634

$\sigma = 0.5$ criterion. It is not necessary that the assumed criterion may show ACS's best performance. Therefore, this instigated the study of performance behavior of ACS method over different levels of parameter σ (see Table 6).

Comparison with SVM Based Method

Since SVM has no concept of PEs, the best of the average accuracy of SVM (average of 100 Monte Carlo simulations for a given pair of c and γ) over the exponential grid space of c and γ is used to compare with the accuracy of the proposed algorithms. Figure 3a shows the topology of performance accuracy over the grid, and Fig. 3b shows the topology of number of support vectors for PID data. Correspondingly, Figs. 4 and 5 show the same for BLD and WBC data, respectively. The maximum testing accuracy that is obtained for PID data from the grid search is 77.2%. Similarly, for BLD and WBC it is 71.4% and 97.07%, respectively.

It would be unfair to directly compare the best accuracy of SVM with the accuracy of the proposed ANN-based algorithms, due to following reason: While calculating the best accuracy of SVM-based method, a fine grid search (exhaustive in nature) over the parameters c and γ is conducted. The possibility to conduct such exhaustive searches over the parameter space is credited to the existence of fast quadratic optimization algorithms like sequential minimal optimization [8]. However, such fine exhaustive grid search for the proposed methods is yet computationally expensive in the case of ANNs (for example, an exhaustive grid search for ACS will require a search over three parameters namely: *number of epochs*, σ, and *number of PEs* in the hidden layer).

However, in order to see the behavior of the ACS algorithm with various levels of σ, a coarse grid search with few grid points have been conducted. The result of this grid search is shown in Table 6. Although the grid is confined to very few

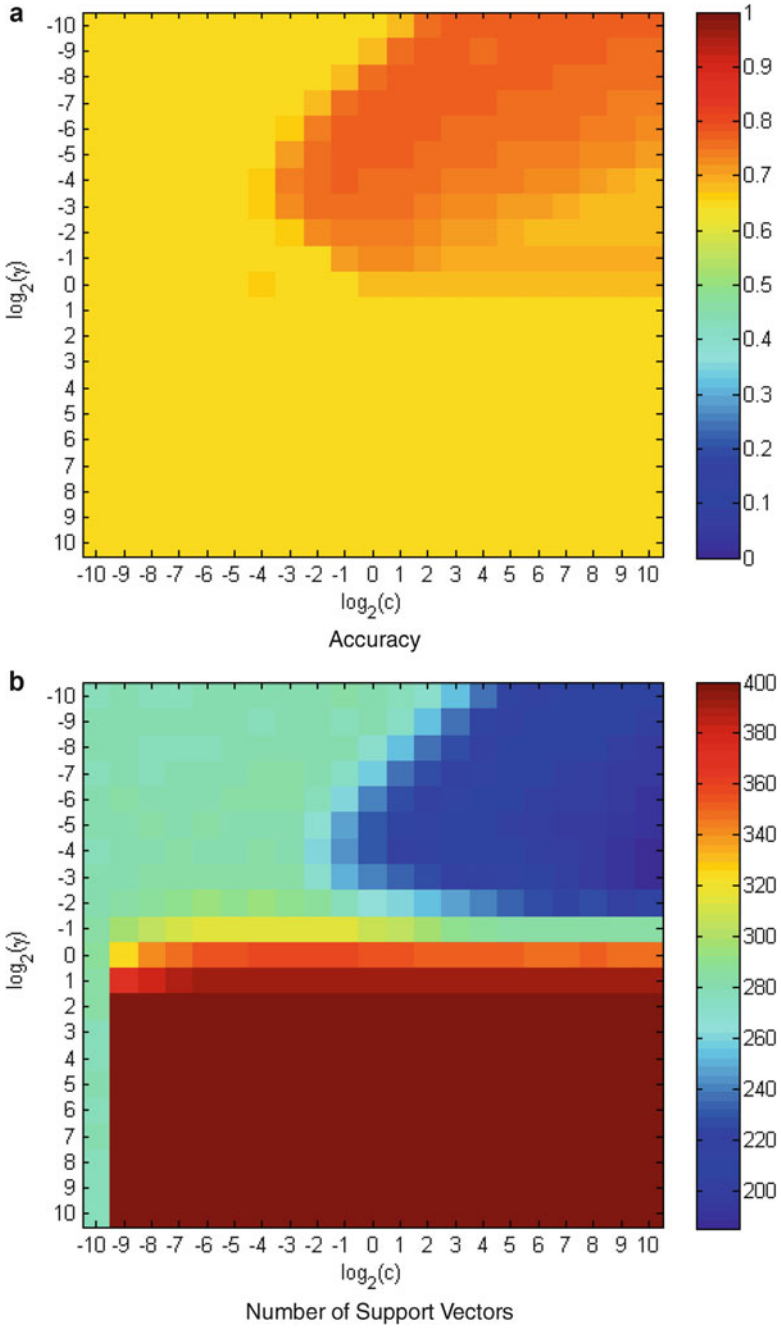

Fig. 3 Performance of SVM on PID data

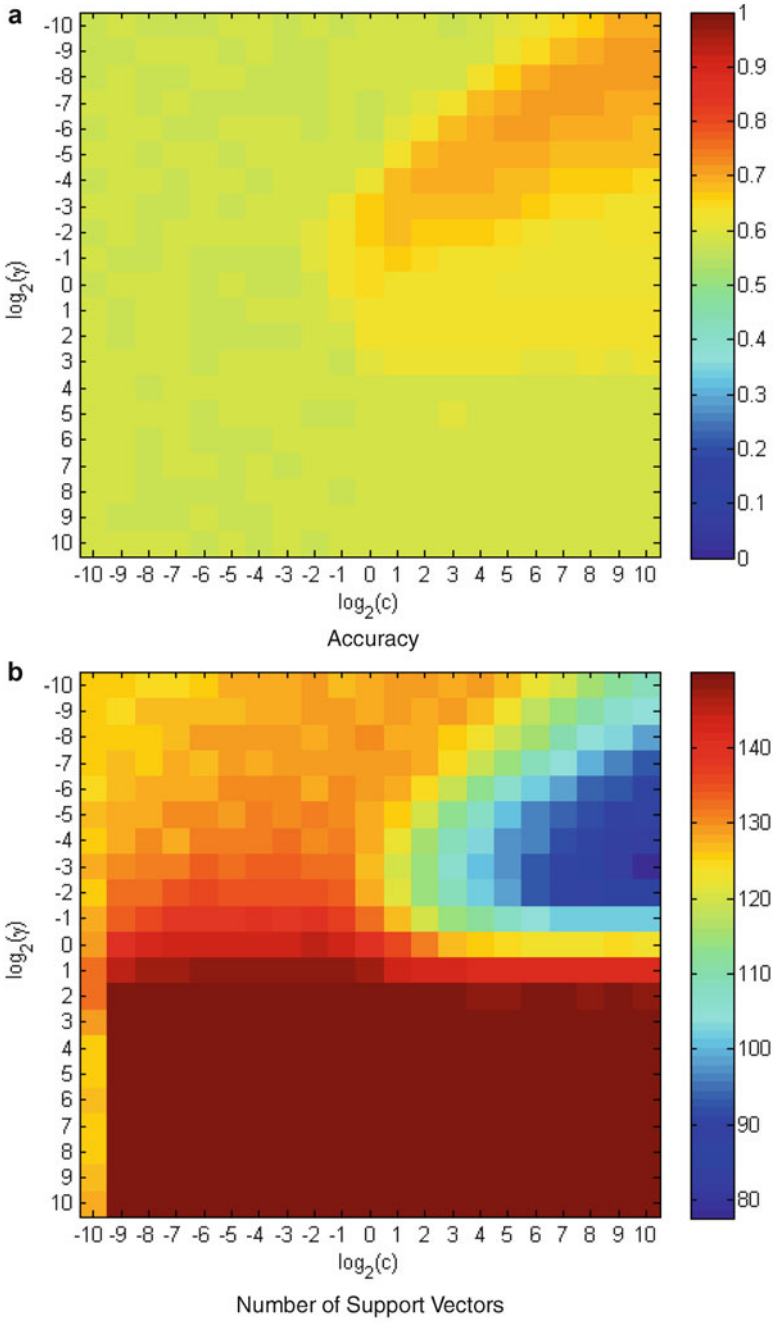

Fig. 4 Performance of SVM on BLD data

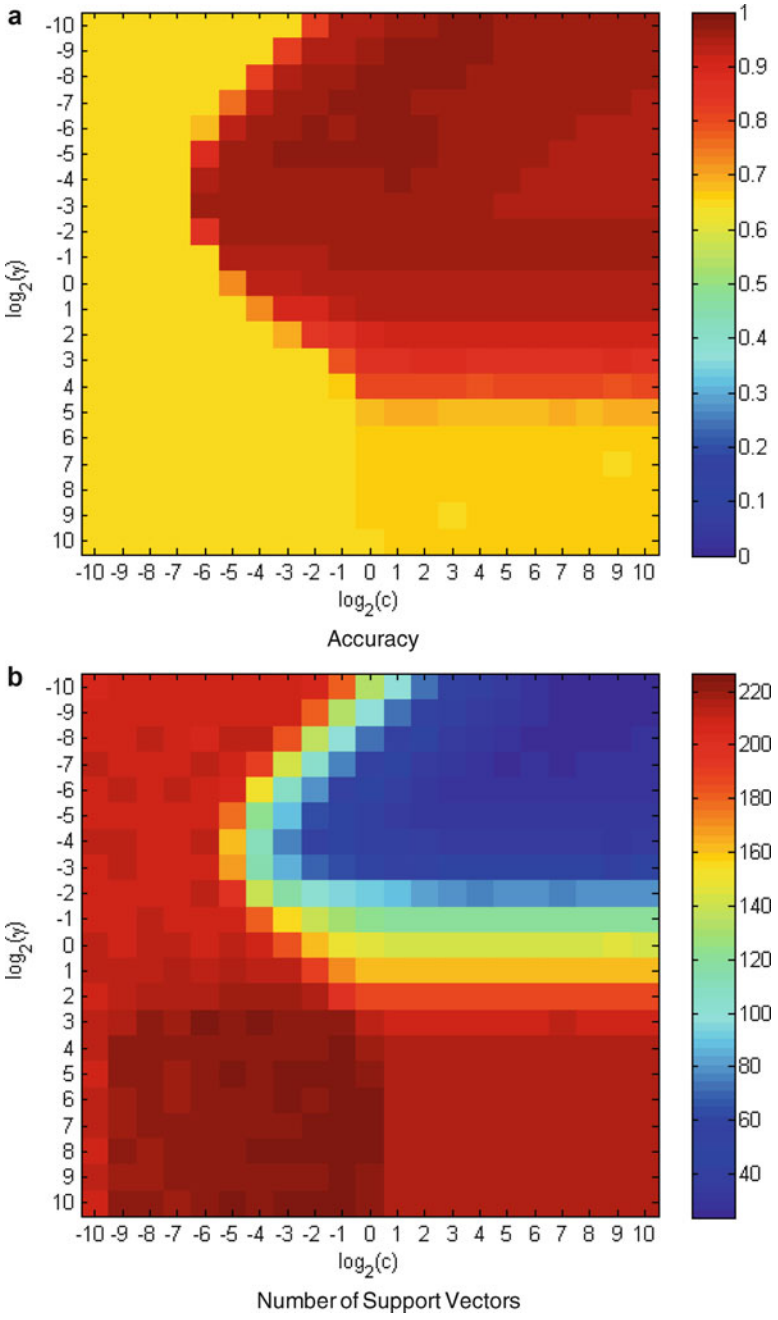

Fig. 5 Performance of SVM on WBC data

grid points, it can be seen that the performance accuracy of ACS algorithm varies with the change of parameters (σ and number of PEs in hidden layer). The results from the grid search show that the performance accuracy of ACS (even with limited PEs and confined levels of σ) is very closer to the best performance accuracy of soft margin kernel-based SVM. Furthermore, even with the limitations (number of PEs in the hidden layer, and number of epochs) ACC beats the best performance accuracy of SVM for WBC data. In addition to that, its performance is very close to that of best SVM performance for the other two data sets.

5 Concluding Remarks

In this chapter two novel approaches that integrate the concepts of correntropy in data classification are proposed. The rationale behind proposing correntropic loss function in data classification is its ability to deemphasize noise (outliers) during the learning phase. Thus noise (or outliers) will not have influence while obtaining classification rule. This is an important property of correntropy function, that can be used in real-world data classification problems. In addition to that, the use of correntropic loss function in two different forms has been illustrated. In type 1 form, the kernel width is allowed to vary in the learning phase. In order to incorporate varying kernel width, a CS-based ANN learning is proposed (ACC method). The ACC method uses the simple well-known delta rule to update the weights. However, the purpose of using this back-propagation mechanism was to illustrate the use of CS-based ANN learning. Different sophisticated methods to replace the back propagation can be used to enhance the basic ACC algorithm.

Furthermore, type 2 form of correntropic loss function has a fixed kernel width. Depending upon the kernel width, the loss function may or may not be convex. Therefore, any classical gradient descent algorithm in ANN framework may converge to a local minimum. To avoid such local convergence, the gradient descent method has been replaced by SA algorithm. Although a simple SA is used within ANN framework, nevertheless, this method can suitably incorporate other specialized forms of SA.

Experiments show that the proposed correntropic loss function improves the classification accuracy of ANN-based classifiers. Furthermore, experiments show that the proposed approaches provide a tough competition to the state-of-the art SVM-based classifier. It can be proposed that the correntopic loss function is a substantial contender for a robust measure in the risk minimization. Moreover, the development of efficient algorithms for the parameter searches in ANNs will further enhance the importance of correntropic loss function.

To conclude, similar to the generalization of SVMs from the basic formulation to the kernel-based soft margin formulation, correntropy-based ANNs can be viewed as a generalized form of ANNs (both in regression [22] and classification). From rigorous experimental results, the usability of correntropy-based ANNs in real-world data classification problems is shown in this chapter.

Acknowledgements This work is partially supported by DTRA and NSF grants.

References

1. Alizamir, S., Rebennack, S., Pardalos, P.M.: Improving the neighborhood selection strategy in simulated annealing using the optimal stopping problem. In: Tan, C.M. (ed.) Simulated Annealing. Springer, New York, pp. 63–382 (2008)
2. Anthony, M., Bartlett, P.L.: Neural Network Learning: Theoretical Foundations. Cambridge University Press, UK (2009)
3. Antonov, G.E., Katkovnik, V.J.: Generalization of the concept of statistical gradient. Avtomat. i Vycisl. Tehn. (Riga) **4**, 25–30 (1972)
4. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory And Algorithms. Wiley, New York (2006)
5. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152. ACM (1992)
6. Catoni, O.: Metropolis, simulated annealing and IET algorithms: Theory and experiments. J. Complex. **12**, 595–623 (1996)
7. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
8. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training support vector machines. J. Mach. Learn. Res. **6**, 1889–1918 (2005)
9. Gunn, S.R.: Support vector machines for classification and regression. ISIS technical report, 14 (1998)
10. Heisele, B., Ho, P., Poggio, T.: Face recognition with support vector machines: Global versus component-based approach. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 688–694. IEEE (2001)
11. Hornick, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
12. Kim, K.I., Jung, K., Park, S.H., Kim, H.J.: Support vector machines for texture classification. IEEE Trans. Pattern Anal. Mach. Intell. **24**(11), 1542–1550 (2002)
13. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671 (1983)
14. Lundy, M., Mees, A.: Convergence of an annealing algorithm. Math. Progr. **34**(1), 111–124 (1986)
15. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biol. **5**(4), 115–133 (1943)
16. Mehrotra, K., Mohan, C.K., Ranka, S.: Elements of Artificial Neural Networks. MIT Press, Cambridge (1997)
17. Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics. Springer, New York (2004)
18. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood, New York (1994)
19. Minsky, M., Seymour, P.: Perceptrons. MIT Press, Cambridge (1969)
20. Pardalos, P., Pitsoulis, L., Mavridou, T., Resende, M.: Parallel search for combinatorial optimization: Genetic algorithms, simulated annealing, tabu search and grasp. Parallel Algorithms for Irregularly Structured Problems, pp. 317–331. Wiley, Hoboken (1995)
21. Pardalos, P.M., Boginski, V.L., Vazacopoulos, A.: Data Mining in Biomedicine. Springer, New York (2007)
22. Principe, J.C.: Information Theoretic Learning: Renyi's Entropy And Kernel Perspectives. Springer, New York (2010)
23. Reeves, C.R.: Modern heuristic techniques for combinatorial problems. Wiley, New York (1993)

24. Robbins, H., Monro, S.L.: A stochastic approximation method. Ann. Math. Stat. **22**, 400–407 (1951)
25. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**(6), 386 (1958)
26. Rubinstein, R.Y.: Smoothed functionals in stochastic optimization. Math. Oper. Res., 26–33 (1983)
27. Santamaría, I., Pokharel, P.P., Principe, J.C.: Generalized correlation function: Definition, properties, and application to blind equalization. IEEE Trans. Signal Process. **54**(6), 2187–2197 (2006)
28. Schölkopf, B., Burges, C., Vapnik, V.: Extracting support data for a given task. In: Proceedings, First International Conference on Knowledge Discovery & Data Mining, pp. 252–257. AAAI Press, Menlo Park (1995)
29. Singh, A., Principe, J.C.: A loss function for classification based on a robust similarity metric. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2010).
30. Styblinski, M.A., Tang, T.S.: Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing. Neural Netw. **3**(4), 467–483 (1990)
31. Syed, M.N., Pardalos, P.M.: Neural network models in combinatorial optimization. Handbook of Combinatorial Optimization. In press
32. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. **2**, 45–66 (2002)
33. Vapnik, V., Golowich, S.E., Smola, A.: Support vector method for function approximation, regression estimation, and signal processing. Adv. Neural Inf. Process. Syst. **9**, 281–287 (1996)
34. Vapnik, V.N.: An overview of statistical learning theory. IEEE Trans. Neural Netw. **10**(5), 988–999 (1999)
35. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (2000)
36. Weston, J., Watkins, C.: Multi-class support vector machines. Technical report, Technical Report CSD-TR-98-04, Department of Computer Science, University of London, Royal Holloway (1998)
37. Zhang, J., Xanthopoulos, P., Chien, J., Tomaino, V., Pardalos, P.M.: Minimum prediction error models and causal relations between multiple time series. In: Cochran, J.J. (ed.) Wiley Encyclopedia of Operations Research and Management Science **3**, 1843–1850 (2011)

Part II
Dynamics of Information in Distributed and Networked Systems

Algorithms for Finding Diameter-constrained Graphs with Maximum Algebraic Connectivity

Harsha Nagarajan, Sivakumar Rathinam, Swaroop Darbha, and Kumbakonam Rajagopal

Abstract This article addresses a problem of synthesizing robust networks in the presence of constraints which limit the maximum number of links that connect any two nodes in the network. This problem arises in surveillance and monitoring applications where wireless sensor networks have to be deployed to collect and exchange time-sensitive information among the vehicles. This network synthesis problem is formulated as a mixed-integer, semi-definite program, and an algorithm for finding the optimal solution is developed based on cutting plane and bisection methods. Computational results are presented to corroborate the performance of the proposed algorithm.

Keywords Algebraic connectivity • Robust network • Mixed-integer Semi-definite program • Unmanned vehicles

1 Introduction

The article addresses a problem of synthesizing robust networks in the presence of diameter constraints. This problem is motivated by surveillance and monitoring applications where mobile adhoc networks (MANETs) [1, 5, 8, 16] have to be deployed to collect and exchange time-sensitive information among the vehicles in the network. Each vehicle serves the role of a node and a communication link between the vehicles serves as an edge (or a link) connecting the nodes. Since there is a cost associated with maintaining a communication link between vehicles, not

H. Nagarajan (✉) • S. Rathinam • S. Darbha • K.R. Rajagopal
Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA
e-mail: harsha_n@tamu.edu; srathinam@tamu.edu; dswaroop@tamu.edu; krajagopal@tamu.edu

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_6,

every pair of vehicles may be in direct communication with each other. As a result, a delay is incurred in communicating the information between any given pair of vehicles and it depends on the number of communication links present in the shortest path between them. The diameter of the network indicates the maximum number of edges in the shortest path among all pairs of nodes and the need for time-sensitive exchange of information among vehicles in a MANET may be better addressed by constraining the diameter of the network.

Since not every pair of vehicles may be "connected" directly by a communication link, one must address the notion of how "well connected" the MANET is. The well connectedness of a network depends on the strength of its communication links as well as its topology. The strength of a communication link is a function of the probability of successful transmission of a packet across the link [6].

Topology of a network affects well connectedness and it specifies which vehicles are in direct communication. Algebraic connectivity is a measure of well connectedness and may be used to differentiate between different network topologies. It is known that algebraic connectivity as a measure of well connectedness is superior to other measures such as the node or the link connectivity of a network [14]; for example, any (unweighted) spanning tree has a node or a link connectivity of one (i.e., it takes only one node or an edge to be removed in order to disconnect the network). On the other hand, it is known that a star network has a higher algebraic connectivity compared to any (Hamiltonian) path in the network. A star network, for instance, is considered to be more robust against a random removal of a node in the network as opposed to a path which gets disconnected upon the removal of any intermediate node [14]. In the case of MANETs, a node may be disconnected if a vehicle is malfunctioning and an edge may be disconnected if the power associated with transmitting information from the originating node to the destined node is not sufficiently large. In the context of graph theory, algebraic connectivity provides a measure of connectedness of any subset of vertices to the remaining graph. With this measure, a subset of vertices is considered to be weakly connected if a normalized cut of the subset (sum of the number of edges leaving the subset) has a low value. Essentially, a tightly connected network with a larger normalized cut corresponds to a network with a higher algebraic connectivity. Additionally, it is well known that the algebraic connectivity of a network determines the rate of convergence of any consensus protocols [2] used by the vehicles to communicate through the links in MANET applications. Since one incurs a cost in maintaining a communication link between vehicles, a natural problem is to consider the problem of maximizing algebraic connectivity of the network subject to an upper bound on the number of communication links that may be maintained.

In the context of MANETS, this article addresses the following network synthesis problem: Given a set of vehicles and the probability of successful transmission of a packet between any two nodes, find a topology of the communication network which links all the nodes such that the diameter of the network is less than a given integer and the algebraic connectivity of the communication network is maximized.

This network synthesis problem is a difficult optimization problem. This problem without the diameter constraints has been shown to be NP-hard by Mosk-Aoyama

in [12]. While existing heuristics such as those in [3, 5, 8, 15] may be used to solve some special cases of the problem considered here, a systematic procedure for finding an optimal solution to the problem is still lacking. The network synthesis problem with the diameter constraints can be posed as a mixed-integer, semi-definite program; however, a direct implementation of this formulation based on a network flow model in MATLAB[1] using open source solvers such as YALMIP [10] and SEDUMI [13] could not find optimal solutions even for a problem with six nodes. In this article, we formulate the network synthesis problem with diameter constraints as a mixed-integer, semi-definite program, and provide an optimal algorithm based on cutting plane and bisection methods to solve the problem. Given the difficulty in finding the optimal solution, we were able to find optimal solutions for instances with 8 nodes in a reasonable amount of time ($\approx$10 min on an average per instance) with the proposed algorithm.

2 Problem Formulation

Let (V, E) represent a graph with $E_0 \subset E$ being the set of edges already in the network. Let w_e represent the edge weight associated with the edge $e \in E$. As discussed in the introduction, in the context of MANET applications, w_e represents the strength of the communication link e. If A, B are two sets, define $A - B := \{x : x \in A, x \notin B\}$. For any edge $e \in E - E_0$, let x_e represent a binary variable that specifies if edge e is chosen or not in the augmented graph; $x_e = 1$ if e is chosen and $x_e = 0$ otherwise. Let the incidence vector, $x \in \{0, 1\}^{|E-E_0|}$, denote the state of the network, with the component x_e corresponding to $e \in E - E_0$. Let e_i denote the ith column of the identity matrix I_n of size $|V| = n$. If the nodes i and j are connected using edge e, let $L_e = w_e(e_i - e_j) \otimes (e_i - e_j)$ where $\otimes$ represents the tensor product. Using this notation, the Laplacian of the augmented structure may be defined as follows:

$$L(x) = \underbrace{\sum_{e \in E_0} L_e}_{L_0} + \sum_{e \in E - E_0} x_e L_e.$$

Let $\lambda_2(L(x))$ denote the algebraic connectivity of the network. The problem of choosing at most q edges from $E - E_0$ so that the algebraic connectivity of the augmented network is maximized and the diameter of the network is within a given constant (D) can be posed as follows:

$$\gamma^* = \max_{x \in \{0,1\}^{|E-E_0|}} \lambda_2(L(x)),$$

[1] The MATLAB program crashed when the diameter constraints were included.

subject to

$$\sum_{e \in E - E_0} x_e \leq q, \tag{1a}$$

$$x \in \{0,1\}^{|E-E_0|}, \tag{1b}$$

$$\delta_{uv}(x) \leq D \quad \forall u, v \in V, \tag{1c}$$

where $\delta_{uv}(x)$ represents the number of edges on the shortest path joining the two nodes u and v in the network with an incident vector x. As stated, there are two challenges that need to be overcome before one can pose the above problem as a mixed-integer, semi-definite program (MISDP). First, the objective is a nonlinear function of the edges in the network; secondly, the diameter constraint as stated in (1c) requires one to implicitly compute the number of edges in the shortest path joining any two vertices. In the following discussion, we explain how we overcome these difficulties, and formulate the network synthesis problem as an MISDP. We also provide a proof of correctness of the formulation at the end of this section.

To address the nonlinear objective, consider the Laplacian matrix, $L(x) = L_0 + \sum_{e \in E - E_0} x_e L_e$. It is well known that the Laplacian is a positive, semi-definite matrix and the smallest eigenvalue of the Laplacian is zero. Let the eigenvalues of $L(x)$ be denoted as $(0 = \lambda_1(L(x))) \leq \lambda_2(L(x)) \leq \ldots \leq \lambda_n(L(x))$. The Laplacian admits a spectral decomposition of the form

$$L(x) = \sum_{i=0}^{n-1} \lambda_{i+1}(L(x))\, e_i \otimes e_i,$$

where e_i is the normalized eigenvector corresponding to eigenvalue, $\lambda_{i+1}(L(x))$. Since $\lambda_1(L(x)) = 0$, the spectral decomposition reduces to

$$L(x) = \sum_{i=1}^{n-1} \lambda_{i+1}(L(x))\, e_i \otimes e_i,$$

or

$$L(x) + \lambda_2(L(x))\, e_0 \otimes e_0 = \lambda_2(L(x))\, e_0 \otimes e_0 + \sum_{i=1}^{n-1} \lambda_{i+1}(L(x))\, e_i \otimes e_i.$$

We will represent $A \succeq B$ if and only if $A - B$ is a positive semi-definite matrix. Clearly, any positive semi-definite matrix A may be represented as $A \succeq 0$. Since $\lambda_i(L(x)) \geq \lambda_2(L(x)) \; \forall \; i \geq 2$, the above equation can be simplified to the following inequality:

$$L(x) + \lambda_2(L(x))\, e_0 \otimes e_0 \succeq \lambda_2(L(x)) \underbrace{\left(\sum_{i=0}^{n-1} (e_i \otimes e_i)\right)}_{I_n},$$

$$\implies L(x) \succeq \lambda_2(L(x))\, (I_n - e_0 \otimes e_0).$$

Therefore, for any connected network x, the Laplacian $L(x)$ and the second eigenvalue of the Laplacian, $\gamma = \lambda_2(L(x))$, satisfy the constraint $L(x) \succeq \gamma\ (I_n - e_0 \otimes e_0)$. Using this relation, we reformulate the equations in (1) as: (the *correctness* of the formulation will be shown later)

$$\gamma^* = \max_{x \in \{0,1\}^{|E-E_0|}} \gamma,$$

subject to

$$L(x) \succeq \gamma\ (I_n - e_0 \otimes e_0), \tag{2a}$$

$$\sum_{e \in E - E_0} x_e \leq q, \tag{2b}$$

$$x \in \{0, 1\}^{|E-E_0|}, \tag{2c}$$

$$\delta_{uv}(x) \leq D \quad \forall u, v \in V. \tag{2d}$$

The next difficulty one needs to address stems from the diameter constraints formulated in (2d). To simplify the presentation, let us limit our search of an optimal network to the set of all the spanning trees. Also, let the parameter D which limits the diameter of the network be an even number. Then, it is well known [4] that a spanning tree has a diameter no more than an even integer (D) if and only if there exists a central node p such that the path from p to any other node in the graph consists of at most $D/2$ edges. If the central node p is given, then one can use the multi-commodity flow formulation [11] to keep track of the number of edges present in any path originating from node p. However, since p is not known a priori, a common way to address this issue is to augment the network with a source node (s) and connect this source node to each of the remaining vertices in the network with an edge (refer to Fig. 1). If one were to find a spanning tree in this augmented network such that there is only one edge incident on the source node and the path from the source node to any other node in the graph consists of at most $\frac{D}{2} + 1$ edges, the diameter constraints for the original network will be naturally satisfied.

In order to impose the diameter constraints formulated in (2d), we add a source node (s) to the graph (V, E) and add an edge joining s to each vertex in V, i.e., $\tilde{V} = V \cup \{s\}$ and $\tilde{E} = E \cup (s, j) \forall\, j \in V$. We then construct a tree spanning all the nodes in $\tilde{V}$ while restricting the length of the path from s to any other node in $\tilde{V}$.

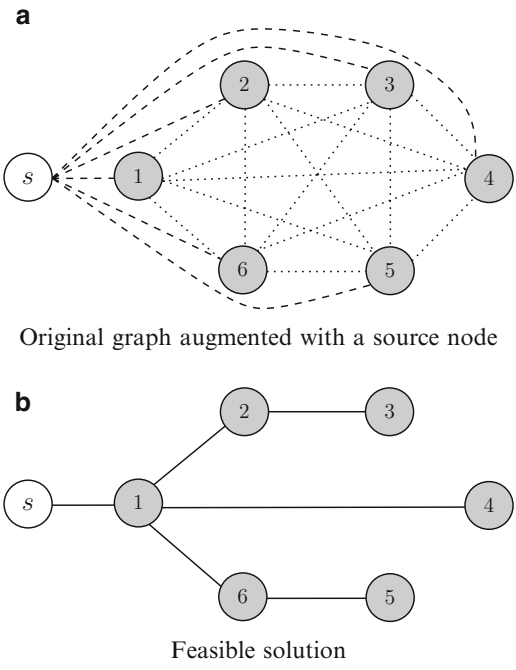

Fig. 1 Illustration of an addition of the source node (s) to the original (complete) graph represented by shaded nodes as show in **(a)**. If one were given that the diameter of the original graph must be at most $D = 4$, then restricting the length of each of the paths from the source node to $(D/2) + 1 = 3$, and allowing only one incident edge on s will suffice as shown in **(b)**

The additional edges emanating from the source node are used only to formulate the diameter constraints, and they do not play any role in determining the algebraic connectivity of the original graph.

Constraints representing a spanning tree are commonly formulated in the literature using the multi-commodity flow formulation. In this formulation, a spanning tree is viewed as a network which facilitates the flow of a unit of commodity from the source node to each of the remaining vertices in $\tilde{V}$. A commodity can flow directly between two nodes if there is an edge connecting the two nodes in the network. Similarly, a commodity can flow from the source node to a vertex v if there is a path joining the source node to vertex v in the network. One can guarantee that all the vertices in V are connected to the source node by constructing a network that allows for a distinct unit of commodity to be shipped from the source node to each vertex in V. To formalize this further, let a distinct unit of commodity (also referred to as the k^{th} commodity) corresponding to the kth vertex be shipped from the source node. Let f_{ij}^k be the kth commodity flowing from node i to node j. Then, the constraints which express the flow of the commodities from the source node to the vertices can be formulated as follows:

$$\sum_{j \in \tilde{V} \setminus \{s\}} (f_{ij}^k - f_{ji}^k) = 1 \quad \forall k \in V \text{ and } i = s, \tag{3a}$$

$$\sum_{j \in \tilde{V}} (f_{ij}^k - f_{ji}^k) = 0 \quad \forall i, k \in V \text{ and } i \neq k, \tag{3b}$$

$$\sum_{j \in \tilde{V}} (f_{ij}^k - f_{ji}^k) = -1 \ \ \forall i, k \in V \text{ and } i = k, \tag{3c}$$

$$f_{ij}^k + f_{ji}^k \leq x_e \ \ \forall\, e := (i, j) \in \tilde{E}, \forall k \in V, \tag{3d}$$

$$\sum_{e \in \tilde{E}} x_e = |\tilde{V}| - 1, \tag{3e}$$

$$0 \leq \ f_{ij}^k \ \leq 1 \ \ \forall i, j \in \tilde{V}, \forall k \in V, \tag{3f}$$

$$x_e \in \{0, 1\} \ \ \forall e \in \tilde{E} \setminus E_0, \quad x_e = 1 \ \ \forall e \in E_0. \tag{3g}$$

Constraints (3a) through (3c) state that each commodity must originate at the source node and terminate at its corresponding vertex. Equation (3d) states that the flow of commodities between two vertices is possible only if there is an edge joining the two vertices. Constraint (3e) ensures that the number of edges in the chosen network corresponds to that of a spanning tree. An advantage of using this formulation is that one now has access directly to the number of edges on the path joining the source node to any vertex in the graph. That is, $\sum_{(i,j) \in \tilde{E}} f_{ij}^k$ denotes the length of the path from s to k. Therefore, the diameter constraints now can be expressed as

$$\sum_{(i,j) \in \tilde{E}} f_{ij}^k \leq (D/2 + 1) \ \ \forall k \in V, \tag{4a}$$

$$\sum_{j \in V} x_{sj} = 1. \tag{4b}$$

To summarize, the mixed-integer, semi-definite program ($\mathcal{F}_1$) for the network synthesis problem is: max γ, subject to the constraints in (2a), (3), and (4). Note that the formulation $\mathcal{F}_1$ is for the case when the desired network is a spanning tree and the bound on the diameter of the spanning tree is an even number. Using the results in [4], similar formulations can also be stated for more general networks with no restrictions on the parity of the bound. In this article, we will concentrate on the formulation presented in $\mathcal{F}_1$.

Lemma 1. *Let an optimal solution corresponding to the formulation $\mathcal{F}_1$ be γ^* and x^*. Then, x^* is a network that solves the network synthesis problem to optimality with γ^* being the algebraic connectivity of x^*.*

Proof. x^* is a spanning tree satisfying the diameter constraints. Now, to show that $\gamma^* = \lambda_2(L(x^*))$, it is enough to prove that $\gamma^* \geq \lambda_2(L(x^*))$ and $\gamma^* \leq \lambda_2(L(x^*))$.

Proof for $\gamma^* \geq \lambda_2(L(x^*))$: We know that x^* is a feasible solution to formulation $\mathcal{F}_1$ with second eigenvalue, $\lambda_2(L(x^*))$. Since this is a maximization problem, γ^* has to be an upper bound on $\lambda_2(L(x))$ for all possible feasible solutions. Hence, $\gamma^* \geq \lambda_2(L(x^*))$.

Proof for $\gamma^* \leq \lambda_2(L(x^*))$: Since (x^*, γ^*) is a feasible solution, we have

$$L(x^*) \succeq \gamma^*(I_n - e_0 \otimes e_0). \tag{5}$$

Let $\hat{v}$ be any unit vector perpendicular to e_0. Then

$$(\hat{v} \cdot L(x^*)\hat{v}) \geq \gamma^*. \tag{6}$$

Hence, from the Rayleigh quotient characterization of the second eigenvalue, it follows that $\lambda_2(L(x^*)) \geq \gamma^*$. □

3 An Algorithm for Computing Optimal Solutions

The proposed formulation with the semi-definite, flow and integer constraints can be solved by YALMIP [10] and SEDUMI [13] which are state-of-the-art, mixed-integer, semi-definite programming solvers widely used by the researchers. However, these solvers could not solve the proposed formulation even for a network involving 6 nodes. Matlab crashed if the formulation included both the semi-definite and the flow constraints. Therefore, we adopt a different approach for finding an optimal solution in this article by casting the algebraic connectivity problem as the following decision problem: Is there an augmented structure with at most q edges from $E - E_0$ such that the algebraic connectivity of the structure is at least equal to a pre-specified value and the diameter of the graph is at most equal to D? One of the advantages of posing this question is that the resulting problem turns out to be a binary semi-definite problem (BSDP) and correspondingly, the tools associated with construction of valid inequalities are more abundant when compared to mixed-integer programs. Also, with further relaxation of the semi-definite constraint, it can be solved using CPLEX, a high-performance solver for integer linear programs.

3.1 Proposed Formulation as a Binary Semi-definite Programming Problem

The decision problem can be mathematically formulated as follows: Is there an x such that

$$L_0 + \sum_{e \in E - E_0} x_e L_e \succeq \lambda_2(I_n - e_0 \otimes e_0), \tag{7a}$$

$$\text{when } x \text{ satisfies the constraints in (3) and (4)?} \tag{7b}$$

The above problem can be posed as a BSDP by marking any vertex in V as a root vertex, r, and then choosing to find a feasible tree that minimizes the degree of this root vertex.[2] In this formulation, the only decision variables would be the binary variables denoted by x_e and the flow variables denoted by f_{ij}^k. Therefore, the BSDP we have is the following:

$$\min \sum_{e \in \delta(r)} x_e,$$

subject to

$$L_0 + \sum_{e \in E - E_0} x_e L_e \succeq \lambda_2 (I_n - e_0 \otimes e_0), \tag{8a}$$

$$x \text{ satisfies the constraints in (3) and (4)}, \tag{8b}$$

where, $\delta(r)$ denotes a cutset defined as $\delta(r) = \{e = (r, j) : \ j \in V \setminus r\}$. If we can solve this BSDP efficiently, then we can use a bisection algorithm to find an optimal structure that will maximize the algebraic connectivity. Now, to solve the BSDP using CPLEX, we do the following: we first do the linear programming relaxation of the semi-definite constraint [9] by taking a finite subset of the infinite number of linear constraints from the semi-infinite program. However, if the solution to the relaxed binary linear program does not satisfy the semi-definite constraint, we add a cut (valid inequality) that ensures that this undesirable solution will not be chosen again and then solve the augmented binary linear program again. The idea of this cutting plane procedure is to construct successively tighter polyhedral approximations of the feasible set corresponding to the desired level of algebraic connectivity. The pseudo code of this procedure is outlined in Algorithm 3. Finding a cut that removes the infeasible solution at each iteration is quite straightforward: If x is the solution to the relaxed program which does not satisfy the semi-definite constraint in (8a), then one can add the cut, $(v^* \cdot L(x^*) v^*) \ \geq \gamma^* (v^* \cdot (I_n - e_0 \otimes e_0) v^*)$, to the program where v^* is the eigenvector corresponding to a negative eigenvalue of $L(x^*) - \gamma^* (I_n - e_0 \otimes e_0)$.

4 Computational Results

The computational results in this section are based on our proposed Algorithm 3 for the problem of maximizing algebraic connectivity under diameter constraints. The proposed algorithm was implemented in C++ programming language and the

[2]There are several ways to formulate the decision problem as a BSDP. For example, one can also aim to minimize the total weight of the augmented graph defined as $\sum_e w_e x_e$. We chose to minimize the degree of a node as it gave reasonably good computational results.

Algorithm 3 Optimal algorithm for maximizing the algebraic connectivity of a network with diameter constraints

Notation: Let $L(x) = \sum_{\forall (i,j) \in E} x_{ij} L_{ij}$.
Let $\mathfrak{F}$ denote a set of cuts which must be satisfied by any feasible solution

Input: A graph $G = (V, E)$, a weight (w_e) for each edge $e \in E$, a root vertex, r, diameter, D, and a finite number of unit vectors, $v_i, i = 1 \ldots M$

Choose any spanning tree satisfying the diameter constraint as an initial feasible solution, x^*

$\hat{\lambda} \leftarrow \lambda_2(L(x^*))$

loop

$\mathfrak{F} \leftarrow \emptyset$

Solve:

$$\min \sum_{e \in \delta(r)} x_e \ ,$$

subject to

$$(v_i \cdot (L(x))v_i) \ \geq \ \hat{\lambda}(v_i \cdot (I_n - e_0 \otimes e_0)v_i) \quad \forall i = 1, .., M,$$

$x$satisfies multi-commodity flow constraints (3) and diameter constraints (4)

x satisfies the constraints in $\mathfrak{F}$

if the above ILP is infeasible **then**

break loop {x^* is the optimal solution with maximum algebraic connectivity}

else

Let x^* be an optimal solution to the above ILP. Let γ^* be the algebraic connectivity corresponding to x^*.

if $L(x^*) \not\geq \gamma^*(I_n - e_0 \otimes e_0)$ **then**

Find the eigenvector v^* corresponding to a negative eigenvalue of $L(x^*) - \gamma^*(I_n - e_0 \otimes e_0)$.

Augment $\mathfrak{F}$ with a constraint $(v^* \cdot L(x^*)v^*) \ \geq \ \gamma^*(v^* \cdot (I_n - e_0 \otimes e_0)v^*)$.

Go to step 6.

end if

end if

$\hat{\lambda} \leftarrow \hat{\lambda} + \epsilon$ {let ϵ be a small number}

end loop

resulting ILPs were solved using CPLEX 12.2 [7]. Augmented ILPs were solved in CPLEX under branch-and-cut framework with default cuts applied to obtain integer solutions and with remaining options set as default. All computational results in this section were implemented on a Dell Precision T5500 workstation (Intel Xeon E5630 processor @ 2.53 GHz, 12 GB RAM).

As discussed in earlier sections, the semi-definite programming toolboxes in Matlab could not be used to solve the proposed formulation with the semi-definite and flow constraints even for instances with 6 nodes primarily due to the inefficient memory management. But solving the same problem without the diameter (flow) constraints in Matlab seems to take large computational time since the convergence to the optimal solution is pretty slow in this case. However, due to the combinatorial explosion resulting from the increased size of the problem, the proposed algorithm with CPLEX solver could provide optimal solutions in a reasonable amount of run time for instances upto 8 nodes. Since we are interested in spanning trees as feasible solutions, there are $8^6 = 262{,}144$ combinatorial possibilities (for a graph with n nodes, there are n^{n-2} possible feasible solutions).

Table 1 Comparison of computational time (CPU time) of the proposed algorithm for different limits on the diameter of the graph and γ^* is the optimal algebraic connectivity. The algorithm was implemented in CPLEX for instances involving 6 nodes

	Diameter ≤ 4		No diameter constraint	
Instance	γ^*	T_1 (s)	γ^*	T_2 (s)
1	39.352	7	559.539	8
2	39.920	4	546.915	8
3	67.270	6	765.744	6
4	50.262	10	713.925	5
5	31.218	8	569.959	4
6	52.344	8	662.326	7
7	35.513	7	637.331	6
8	38.677	7	704.89	6
9	46.427	11	574.132	5
10	40.945	7	597.241	5
11	36.770	10	586.950	9
12	42.885	6	587.027	5
13	30.880	8	569.482	10
14	47.583	3	543.145	6
15	37.277	4	517.401	9
16	37.439	11	704.228	8
17	51.434	10	639.456	3
18	42.476	3	620.974	10
19	29.934	3	576.275	4
20	46.980	6	536.366	6
21	25.955	6	630.748	9
22	49.220	6	601.309	4
23	53.282	6	607.615	6
24	45.909	5	524.214	6
25	48.120	3	549.210	3

Table 2 Comparison of computational time (CPU time) of the proposed algorithm for different limits on the diameter of the graph and γ^* is the optimal algebraic connectivity. The algorithm was implemented in CPLEX for instances involving 8 nodes

	Diameter ≤ 4		Diameter ≤ 6		No diameter constraint	
Instance	γ^*	T_1 (s)	γ^*	T_2 (s)	γ^*	T_3 (s)
1	66.1636	298.10	93.0846	184.26	631.739	495.23
2	39.2994	477.34	54.3061	416.43	631.883	980.98
3	44.8588	803.45	45.9793	634.54	604.213	4, 253.01
4	66.5337	394.02	78.7357	221.21	757.490	815.01
5	33.8383	519.28	53.8226	480.23	755.205	706.25
6	46.6083	1, 033.09	75.6113	349.12	513.994	586.34
7	51.1379	781.07	63.3915	385.17	550.717	949.30
8	42.8026	931.50	77.4458	319.51	807.108	333.93
9	58.1182	489.43	84.7166	348.82	769.641	482.55
10	50.5110	492.11	54.3155	323.33	646.711	1, 789.64
11	43.6888	791.01	107.1820	212.34	729.171	472.71
12	47.5213	693.13	82.2919	219.20	655.867	1, 061.16
13	42.4918	468.44	53.2514	698.21	698.129	1, 421.38
14	41.1752	445.26	48.9485	261.18	523.118	977.67
15	44.8202	518.13	63.8735	509.77	639.540	504.42
16	40.1853	540.19	72.1540	396.25	690.719	661.91
17	66.6196	480.70	108.0970	254.47	735.361	476.87
18	62.9801	499.78	69.1063	233.33	622.840	1, 372.58
19	40.7602	542.69	54.9466	343.04	650.096	236.65
20	60.1121	607.19	81.2138	209.15	607.008	590.38
21	66.3578	588.31	80.3600	408.78	609.370	730.82
22	42.8765	776.38	75.5561	458.80	666.251	734.43
23	42.7949	400.03	62.8144	638.11	444.903	942.26
24	63.1568	590.91	73.7841	333.03	680.411	804.27
25	31.3830	232.18	44.6972	231.16	630.107	818.93

The weighted adjacency matrix of a complete graph (A_{rand}) for each random instance was generated using $A_{\text{rand}} = (M \circ R) + (M \circ R)^T$ where $\circ$ corresponds to the Hadamard product of magic square (M) and a square matrix (R) with zero diagonal entries whose off-diagonal entries are the pseudorandom values obtained from the standard uniform distribution on an open interval $(0, 1)$. The term $A_{\text{rand}}[i, j]$ corresponds to the strength of the communication link connecting vertices i and j.

We shall now compare the computational times of the proposed algorithm to obtain optimal solutions for different values of the bound on the diameter. The results shown in Tables 1 and 2 are for 25 random instances generated for networks with 6 and 8 nodes, respectively. Based on the results in Table 1, we observed that the average run time for obtaining optimal solution for the 6 nodes problem with diameter constraint was (average T_1) 6.6 s and without diameter constraint was (average T_2) 6.3 s. Based on the results in Table 2, we observed that the average

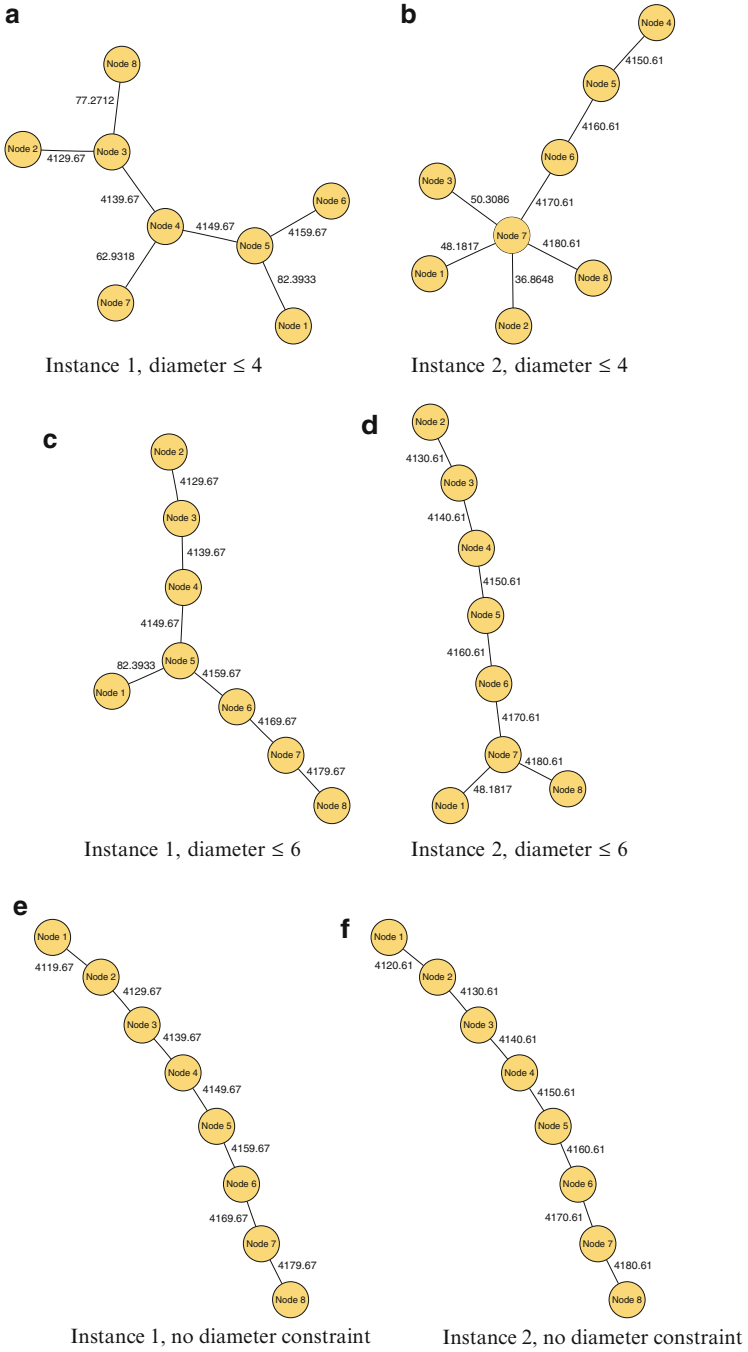

Fig. 2 (**a**, **c**, and **e**) correspond to optimal networks with maximum algebraic connectivity subject to various diameter constraints, for instance, 1 (from Table 2). Similar plots, for instance, 2 are also shown in (**b**, **d**, and **f**)

run time for the problem without diameter constraints (average $T_3 = 927.95$ s) was 1.61 times greater than the average run time for the problem with diameter ≤ 4 (average $T_1 = 575.75$ s) and 2.56 times greater than the average run time for the problem with diameter ≤ 6 (average $T_2 = 362.77$ s). Optimal networks with various diameter constraints corresponding to instances 1 and 2 of Table 2 with 8 nodes may be found in Fig. 2.

5 Conclusions

In this article, we considered the problem of synthesizing networks with maximum algebraic connectivity subject to a constraint on the diameter of a graph and proposed a systematic procedure to solve this problem. Such a problem arises while synthesizing robust communication networks in surveillance and monitoring applications. The article provides a formulation of the network synthesis problem as a mixed-integer, semi-definite, linear program and provides an optimal algorithm based on cutting plane and bisection methods. The algebraic connectivity of a network is posed using a semi-definite constraint and the diameter of the graph is formulated using a multi-commodity flow formulation. We provide preliminary computational results for a 6-node and 8-node problem for varying limits on the diameter of the graph. Even though the proposed work is an improvement over state-of-the-art mixed-integer SDP solvers for the problem of maximizing algebraic connectivity with diameter constraints, there is definitely a need for faster algorithms that can handle more number of vertices.

References

1. Burdakov, O., Doherty, P., Holmberg, K., Kvarnstrom, J., Olsson, P.-R.: Positioning unmanned aerial vehicles as communication relays for surveillance tasks. In: Proceedings of Robotics: Science and Systems, Seattle, USA, June 2009
2. Fax, J. A., Murray, R. M.: Information flow and cooperative control of vehicle formations. IEEE Trans. Autom. Contr. **49**(9), 1465–1476 (2004)
3. Ghosh, A., Boyd, S.: Growing well-connected graphs. Proceedings of the 45th IEEE Conference on Decision and Control, **78**, 6605–6611 (2006)
4. Gouveia, L., Magnanti,T.L.: Network flow models for designing diameter-constrained minimum-spanning and steiner trees. Networks **41**(3), 159–173 (2003)
5. Han, Z., Lee Swindlehurst, A., Ray Liu, K. J.: Optimization of manet connectivity via smart deployment/movement of unmanned air vehicles. IEEE Trans. Veh. Technol. **58**, 3533–3546 (2009)
6. Han, Z., Swindlehurst, A.L., Liu, K.J.R.: Smart deployment/movement of unmanned air vehicle to improve connectivity in manet. In: Wireless Communications and Networking Conference, vol. 1, pp. 252 –257. WCNC (2006)
7. IBM - ILOG: CPLEX optimization studio 12.2. http://www.ilog.com/products/cplex
8. Ibrahim, A.S., Seddik, K.G., Ray Liu, K.J.: Improving connectivity via relays deployment in wireless sensor networks. In: GLOBECOM, pp. 1159–1163 (2007)

9. Krishnan, K.: Linear programming approaches to semidefinite programming problems. PhD thesis, Rensselaer Polytechnic Institute (2002)
10. Lofberg, J.: Yalmip: A toolbox for modeling and optimization in matlab. In: IEEE International Symposium on Computer Aided Control Systems Design, pp. 284–289. IEEE (2004)
11. Magnanti, T.L., Wolsey, L.A.: Optimal trees. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Handbooks in Operations Research and Management Science, vol. 7, pp. 503–615. Springer, New York (1995)
12. Mosk-Aoyama, D.: Maximum algebraic connectivity augmentation is NP-hard. Oper. Res. Lett. **36**(6), 677–679 (2008)
13. Sturm, J.F.: Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optim. Methods softw. **11**(1), 625–653 (1999)
14. Wang, H.: Robustness of Networks. PhD thesis, Delft University of Technology (2009)
15. Wei, P., Sun, D.: Weighted algebraic connectivity: An application to airport transportation network. In: Proceedings of the 18th IFAC World Congress, Milan, Italy (2011)
16. Yu, K., Zhan, P., Swindlehurst, A.L: Wireless relay communications with unmannned aerial vehicles: Performance and optimization. IEEE Trans. Aerosp. Electron. Syst. **47**, 2068–2085 (2010)

Robustness and Strong Attack Tolerance of Low-Diameter Networks

Alexander Veremyev and Vladimir Boginski

Abstract This chapter analyzes optimal attack-tolerant network design and augmentation strategies for bounded-diameter networks. In the definitions of attack tolerance used in this chapter, we generally require that a network has a *guaranteed* ability to maintain *not only the overall connectivity*, but also preserve the same diameter after multiple failures of network components (nodes and/or edges), regardless of whether these failures are random or targeted. This property is referred to as "*strong*" *attack tolerance*, whereas the property of a network to maintain just the regular connectivity after node/edge failures (with no explicit restriction on the diameter), such as in the case of K-connected networks, is referred to as "weak" attack tolerance. We analyze necessary and sufficient conditions for guaranteed "weak" and "strong" attack tolerance properties for fixed-diameter networks, including the important special case of diameter-2 (two-hop) networks. We demonstrate that the recently introduced concept of an *R-robust 2-club* is the *only* diameter-2 network configuration that is guaranteed to have a strong attack tolerance property (i.e., maintain both connectivity and diameter 2) after any $R-1$ edges are deleted. Furthermore, we demonstrate that if all edges have the same construction cost, the problem of optimal R-robust 2-club network design has an *exact analytical solution* that requires $O(Rn)$ constructed edges, which makes this configuration asymptotically as cost-efficient as a regular sparse connected network. We also give linear 0–1 formulations for related network design and augmentation

A. Veremyev
University of Florida, Industrial and Systems Engineering, 303 Weil Hall, Gainesville, FL 32611, USA
e-mail: averemyev@ufl.edu

V. Boginski (✉)
University of Florida, Industrial and Systems Engineering, 1350 N Poquito Rd, Shalimar, FL 32579, USA
e-mail: vb@ufl.edu

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_7,

problems with different edge construction costs, which are NP-hard in the general case. Illustrative examples are provided to demonstrate the considered concepts and results.

Keywords Robust network design • Combinatorial optimization • Strong attack tolerance • k-clubs • R-robust k-clubs

1 Introduction

The task of ensuring efficient and reliable operation networked systems (e.g., communication/information exchange networks) plays an extremely important role in many areas nowadays. *Connectivity* is an essential characteristic of any operational network; however, the definition of connectivity (i.e., the existence of a path between every two nodes) may not provide the required robustness characteristics, since long paths between nodes may make networks rather vulnerable and expensive to operate, especially if every node and/or edge in the path can potentially experience a temporary or permanent failure. The failures of network components can be caused by natural or man-made disruptions (e.g., natural disasters, adversarial attacks, etc.). In general, the failures can be either *random* (no specific network components are targeted; these failures are often referred to as *errors*) or *targeted* (certain critical network components, e.g., high-degree nodes or high-load edges, are targeted; these failures are referred to as *attacks*). In this context, the important issues of *error and attack tolerance* (i.e., the ability of a network to remain operational even in the presence of random or targeted disruptions) need to be efficiently addressed in the design and augmentation of modern network infrastructures.

Many real-world networks follow certain patterns in their degree distributions, and they can often be modeled as *power-law* or *uniform* random graphs. In particular, many publicly available datasets on real-world network infrastructure connectivity patterns (e.g., telecommunications/internet, air transportation, etc.) suggest that the power-law model is applicable to characterizing these networks. Empirical studies of error and attack tolerance of power-law and uniform random graphs have been done in the past (e.g., [2]); however, research on rigorous theoretical justifications and provably optimal strategies for robust network design and augmentation (that would *simultaneously* guarantee error/attack tolerance, low diameter, and cost efficiency) is far from complete.

In many practical applications one needs to consider both *node failures* and *edge failures*. In particular, the "$N-1$" robustness criterion stating that a network should remain operational after a failure of any one edge is used in certain application areas, such as the analysis of power grid robustness. From mathematical perspective, edge failures do not reduce the size (number of nodes) of the residual network; therefore, if a large-scale network is robust with respect to multiple (say up to R) edge failures, it guarantees that none of the nodes in this network would become isolated. This can

be referred to as the "$N - R$" robustness criterion or, in the terminology used later in this chapter, a "weak edge attack tolerance property of level R." In the analysis below, we will address both "edge attack tolerance" and "node attack tolerance" characteristics of different types of networks. In our definitions of attack tolerance, we will require that a network is *guaranteed* to *at least* stay connected after an attack on *any* one or multiple edges/nodes. Furthermore, we will impose more restrictive requirements on network connectivity patterns that will be referred to as "strong attack tolerance."

In this chapter, we will address the aforementioned issues from a rigorous modeling and optimization perspective. In particular, our goal is to develop optimal strategies for network design and augmentation that explicitly take into account certain robustness/attack tolerance criteria (that will be formally defined below), as well as the total cost of constructing a new network or enhancing (upgrading) the existing network.

In a basic network design problem, n nodes need to be optimally connected by a set of arcs/edges so that the total arc/edge construction cost is minimized. Although the construction cost of each possible edge can be different and depends on many practical factors, the total cost will be determined by the *number of constructed edges* under the assumption that all edge construction costs are identical or sufficiently close to each other. Clearly, to ensure the overall connectivity of the constructed network, one needs to construct *at least* $n - 1$ *edges* (here and further we assume a simple undirected graph, i.e., if multiple edges connect the same pair of nodes, then they are represented as a single edge, and all the edges are undirected). Two extreme cases of possible connected network configurations with n nodes and $n - 1$ edges are a "chain" (all nodes connected consecutively) and a "star" (also referred to as a *hub-and-spoke* configuration with one central "hub" node), and any other configuration with $n - 1$ edges would generally be a spanning tree in the considered network. While neither of these configurations is guaranteed to stay connected even after one targeted attack on an appropriate node or edge, the hub-and-spoke configuration (and its modifications) is often used in a variety of applications, since it has the *lowest diameter* among all connected networks with $n - 1$ edges. The *diameter* of a graph (network), which will be formally defined below, is the maximum length shortest path between any two graph vertices. Low diameter (e.g., a small number of intermediary nodes and edges between any pair of nodes) is important for some types of networks (e.g., communication networks); therefore, it can be considered as one of the significant requirements. A network cluster that by definition is explicitly guaranteed to have a diameter of at most k is referred to as a *k-club* [3, 20]. Clearly, any hub-and-spoke structure with n nodes and $n - 1$ edges is a 2-club; therefore, a 2-club can be naturally considered as a cost-efficient connected network structure that also satisfies the low-diameter requirement. Since 2-clubs have attractive properties of both *cost efficiency* and *low diameter*, it is reasonable to attempt to develop a network configuration that preserves these advantages of a 2-club, but at the same time is guaranteed to remain connected (or even to remain a 2-club) if one or multiple nodes and/or edges fail due to attacks or errors. It turns out that such efficient network configurations exist;

moreover, they can be proven to be optimal in terms of necessary and sufficient conditions for satisfying certain robustness and attack tolerance characteristics.

Particularly, we will consider two types of attack tolerance properties, which will be referred to as "*weak*" and "*strong*" attack tolerance. We say that a network (k-club) has a weak attack tolerance property if it always remains *connected* after the deletion of *any* one node or edge, and a network (k-club) has a strong attack tolerant property if it *remains a k-club* after the deletion of *any* one node or edge. We also define *weak/strong attack tolerance of level R* if the aforementioned requirements hold after the deletion of *any R* nodes/edges.

2 Notations and Related Previous Work

Let $G = (V, E)$ be an undirected graph with n nodes. For any pair of nodes (i, j) let $d_G(i, j)$ be the length of a shortest path between nodes i and j, and $d(G) = \max_{i,j \in V} d_G(i, j)$ be the *diameter* of G. We also use $e(G)$ and $v(G)$ as the edge and node connectivity of the graph, respectively, i.e., the minimum number of edges or nodes whose removal disconnects the graph G (that is, breaks the graph into two or more components). The minimum nodal degree of the graph G is $\delta(G)$, and a graph (subgraph) with $\delta(G) \geq k$ is known to be a k-core.

A graph (subgraph) with diameter k is referred to as a *k-club*, a definition originally introduced in the social networks literature [19, 20]. Veremyev and Boginski [25] also proposed a generalization of this concept referred to as an *R-robust k-club* (or, a *(k, R)-club*), which requires the existence of at least R "distinct" paths of length at most k between any pair of nodes. In other existing literature k-clubs are also known as hop-constrained networks, meaning that there is a path between any pair of nodes with at most k hops (edges) [8].

In the definition of "distinct" paths, one can generally require that these paths must be *internally node disjoint*. This requirement can be relaxed by introducing alternative (weaker) requirements, such as: (1) *internally edge-disjoint* paths that may share common nodes; or (2) paths that have a difference in *at least one edge*. For the latter definition, compact linear integer formulations have also been developed in [25] to solve the maximum R-robust k-club problem.

Furthermore, for the special case of an *R-robust 2-club*, all of the above definitions of distinct paths are equivalent. Based on this observation, maximum R-robust 2-clubs in graphs can be found using a compact linear 0–1 formulation proposed in [25]. The key characteristic of R-robust 2-clubs in terms of its attack tolerance properties is the fact that it *not only stays connected* after $R - 1$ nodes and/or edges are destroyed, but it also maintains diameter 2. This fact provides a significant advantage in the applications where the existence of short and reliable paths between any two nodes is of critical importance (i.e., in military communication networks). This also clearly distinguishes the definition of R-robust k-clubs (and R-robust 2-clubs) from the well-known definition of a *K-connected*

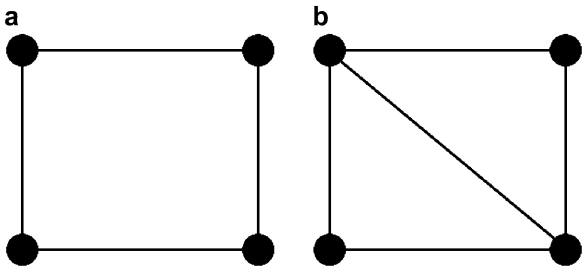

Fig. 1 Illustrative example for $n = 4$: (**a**) a 2-connected graph with diameter 2, which is *not* a 2-robust 2-club; (**b**) a 2-robust 2-club

graph, which also preserves the connectivity after any $K - 1$ network components are destroyed, but it *does not impose any explicit restrictions on the diameter* of the original and the remaining network.

Figures 1a and 1b give a simple illustrative example of the difference between the aforementioned definitions.

Several previous studies attempted to address the issue of constructing a network with small diameter and the existence of disjoint paths between network nodes. In [23] and [24], the authors considered the problems of constructing a K-connected graph with minimum number of edges and "quasiminimal diameter." However, in the definition of "quasiminimal diameter," the diameter of the constructed graph depends on the number of nodes n, whereas in the case of an R-robust k-club, the diameter k does not depend on any other parameters. Moreover, the constructed K-connected graph with quasiminimal diameter does not necessarily preserve the same diameter after the deletion of up to $K - 1$ nodes/edges (although it does preserve the basic connectivity).

In a recent study, Bendali et al. [6] considered the problem of finding a minimum cost *subgraph* of G such that for two *specified* nodes s and t there are at least R edge-disjoint paths of length at most k and obtained the results for certain special cases. They generalized the earlier work of [11] ($k = 2$), and [15] ($R = 2, 3$). A drawback of that problem setup is that the disjoint paths are found only between the *given* two nodes s and t. The generalized version of this problem was considered in [8], where the path requirements were imposed on every pair of nodes from a given set of pairs of nodes $Q \in V \times V$ and the problem was formulated for any R, k. In the problems considered in this chapter, we impose the requirement that *any* two nodes are connected by multiple disjoint paths of length at most 2 ($k = 2$). In [12] the authors present a computational study of the problem with $k = 2$, $R = 1$. It should be pointed out that previous work in this area was primarily concentrated on the computational study of these problems, whereas in this chapter we focus on theoretical analysis of optimal solutions.

A significant amount of work was also previously done on the computational studies of connectivity augmentation problems. Problems of optimally augmenting an existing network in order to make it K-connected (without any restrictions on the path lengths) are discussed in [9, 13, 14]. Recent work for the special case of

$K = 2$ (biconnectivity augmentation) is presented in [5, 17], and [16]. For the *hop-constrained* design and augmentation problem and overview of the previous work, we refer the reader to [8].

In the terminology used in this chapter, we refer to the attack tolerance characteristics identical to those of K-connected graphs as the *"weak" attack tolerance property*, since it only preserves the basic connectivity, but not necessarily preserves the same diameter, whereas the case when a graph not only stays connected, but also preserves the same (ideally, low) diameter, is referred to as the *"strong" attack tolerance property*. These properties are formally defined below.

Definition 1 (weak vs. strong attack tolerance). A connected graph $G = (V, E)$ with diameter k has a weak (strong) edge and/or node attack tolerance property of *level* R if it stays connected (and in the case of strong attack tolerance maintains diameter k) after the deletion of any R edges and/or nodes.

Although this chapter presents linear 0–1 formulations for optimal attack-tolerant 2-hop constrained network design and augmentation, a comparison of the proposed problem formulations to those developed in the previous literature is beyond the scope of the chapter. The emphasis of this study is on obtaining theoretical bounds on optimal solutions of these problems, as well as analytically identifying exact optimal solutions of the considered problems for certain special cases. Numerical experiments will be conducted for illustrative purposes to support theoretical results.

Specifically, we prove several facts on structural properties of weakly and strongly attack-tolerant 2-clubs. We show that any 2-club has a weak (strong) attack tolerance property of level R *if and only if* it is a 2-club/R-core (R-robust 2-club). We also derive sharp lower bounds on the number edges in 2-club/2-cores, and R-robust 2-clubs. We prove that any 2-club/2-core has at least $(3n - 3)/2$ edges, and any R-robust 2-club has at least $Rn - R(R + 1)/2$ edges. Optimal network configurations that contain exactly the lower bound of the number of edges are also identified.

3 Necessary and Sufficient Conditions for Attack Tolerance of k-clubs

In this section, we will analyze attack tolerance properties of k-clubs. Clearly, k-clubs (e.g., 2-clubs) do have attractive properties, such as low diameter and construction cost efficiency, which makes these configurations the subject of special interest in this chapter. In terms of the required number of edges, designing a network with a small diameter (i.e., diameter 2) would require $O(n)$ instead of $O(n^2)$ edges (as in the case of cliques [7], quasi-cliques [1], and k-plexes [4, 22]). However, these structures generally lack attack tolerance properties with respect to targeted node and edge attacks (consider, for instance, a hub-and-spoke configuration, which is an optimal 2-club in terms of the minimal number of constructed edges).

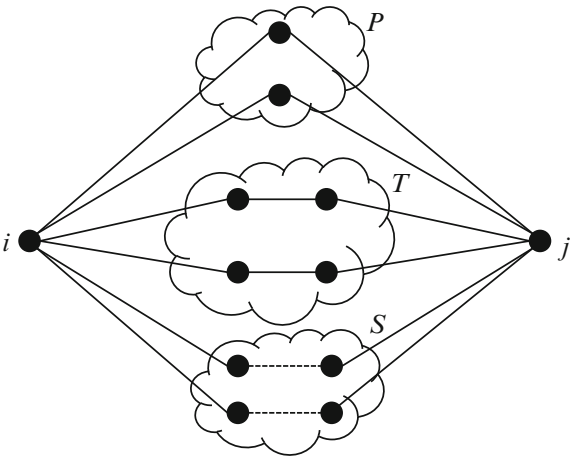

Fig. 2 Illustration of the edge-disjoint paths used in the proof of Proposition 1

Despite the lack of attack tolerance properties in general case, we will show that in certain special cases, weak and strong attack tolerance properties of k-clubs can be theoretically guaranteed. We will formulate sufficient conditions for a k-club to be tolerant to node and edge attacks; moreover, we will show that in the case of a 2-club, *necessary and sufficient* conditions for guaranteed edge attack tolerance can be proven. Specifically, we will show that the aforementioned recently introduced concept of an R-robust k-club can indeed provide at least the sufficient conditions for strong attack tolerance with respect to multiple node/edge failures. Moreover, we will prove that in the case of 2-clubs, an R-robust 2-club is a necessary and sufficient configuration that is guaranteed to be strongly attack tolerant (i.e., maintain the connectivity and diameter 2) after any $R - 1$ edges are destroyed. We will also formulate related necessary and sufficient conditions for weak attack tolerance of 2-clubs, which can be utilized when only the connectivity of the residual network after an attack is the property of interest.

The conditions derived below are easily interpretable and can be viewed as equivalent representations of attack tolerant k-clubs. In case of 2-clubs, they also can be easily incorporated into the optimization problems for designing a minimum-cost weakly or strongly attack-tolerant 2-club.

Proposition 1 (weak attack tolerance requirement for 2-clubs). *Let $G = (V, E)$ be a 2-club, then $e(G) \geq R$* if and only if *$\delta(G) \geq R$ (i.e., G is at least an R-core).*

Proof. The necessary condition follows immediately from the well-known fact from graph theory that $e(G) \geq \delta(G)$.

To prove the sufficient condition, we show that there exist at least R edge-disjoint paths between any pair of nodes (i, j) (Fig. 2). Without loss of generality, we consider the case with $(i, j) \notin E$, and $deg(i) = deg(j) = R$.

Let $N(i)$ be a neighborhood of node i, i.e $N(i) = \{j \in V : (i, j) \in E\}$, and let $P = N(i) \cap N(j)$. From the remaining set of nodes $(N(i) \cup N(j)) \setminus P$ let chose

the maximum set of pairs of nodes $T = \{p \in N(i), q \in N(j)) : (p,q) \in E\}$ so that no pairs have common nodes. Sets P and T can also be viewed as the sets of $|P| + |T|$ edge-disjoint paths between the pair of nodes (i, j).

From the remaining set of nodes in $N(i)$ and $N(j)$, let form a set of arbitrary pairs $S = \{p \in N(i), q \in N(j))$ so that no pairs have common nodes. Note, that none of the remaining nodes in $N(i)$ is directly connected to any of the remaining node in $N(j)$ since T is the maximum set of such pairs. Since G is a 2-club, then any pair of nodes in S is connected through a path of length 2, going through the node that does not belong to S. Hence, all these paths between any pair of nodes in S are edge disjoint. Clearly, they are also edge disjoint with already mentioned $|P| + |T|$ paths. By definition, $|P| + |T| + |S| = N(i) = N(j) = R$, what ends the proof of this proposition. □

Note that in the proof of this proposition we also showed that if $G = (V, E)$ is a 2-club and R-core, then after the deletion of any $R - 1$ edges, it not only remains connected but also becomes a *4-club*, in other words, the residual network still maintains a relatively low diameter, although it increases compared to the original network, so the strong attack tolerance property is formally not satisfied.

Next, we consider the case of strongly node/edge attack-tolerant 2-clubs.

Proposition 2 (strong attack tolerance requirement for 2-clubs). *For any graph* $G = (V, E)$*:*

1. $\forall E_R = \{e_1, \ldots, e_{R-1}\} \subset E$, $G_{E_R} = (V, E \setminus E_R)$ *is a 2-club* if and only if $G = (V, E)$ *is an R-robust 2-club.*
2. $\forall E_l = \{e_1, \ldots, e_l\} \subset E$ *and* $\forall V_m = \{v_1, \ldots, v_m\} \subset V$ *so that* $l + m = R - 1$, $G_{V_m, E_l} = (V \setminus V_m, E \setminus E_l)$ *is a 2-club* if $G = (V, E)$ *is an R-robust 2-club.*

Proof. The proof follows immediately from the definition of an R-robust 2-club. □

The reader might notice that we do not claim that if $\forall v \in V$, $G_v = (V \setminus v, E)$ is a 2-club, then $G = (V, E)$ is a 2-robust 2-club ($l = 0, m = 1, R = 2$ in the second statement). In fact, it is not true in general. Figure 1a shows a 2-club that satisfies this property, but is not a 2-robust 2-club. Note that R-robust 2-clubs provide mixed (edge and/or node) attack tolerance properties.

Remark 1. For any graph $G = (V, E)$: $\forall E_l = \{e_1, \ldots, e_l\} \subset E$ and $\forall V_m = \{v_1, \ldots, v_m\} \subset V$ so that $l + m = R - 1$, $G_{V_m, E_l} = (V \setminus V_m, E \setminus E_l)$ is a k-club *if* $G = (V, E)$ is an R-robust k-club.

This remark shows that R-robust k-clubs ($k > 2$) provide only a *sufficient* condition for strong edge attack tolerance of level $R-1$. Figure 3 shows an example of a 4-club that has a strong edge attack tolerance property, but is not a 2-robust 4-club, demonstrating that the necessary condition does not hold.

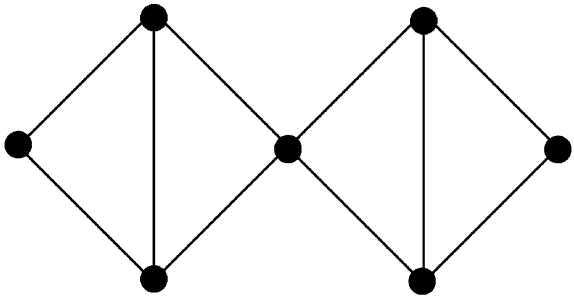

Fig. 3 A strongly edge attack-tolerant 4-club which is not a 2-robust 4-club

4 Linear 0–1 Formulations of Optimal Design and Augmentation Problems for Attack-Tolerant 2-clubs

4.1 Computational Complexity of the Considered Problems

Li et al. [18] and Dahl and Johannessen [12] showed that given a graph $G = (V, E)$ the problem of finding a superset of edges $E' \supseteq E$ such that the graph $G' = (V, E')$ has diameter no greater than 2 (or 2-club) is NP-hard. In the problems considered in this chapter we require the new graph $G' = (V, E')$ not only to have diameter no greater than 2, but also to satisfy other requirements, i.e., the minimum node degree of every node for the R-core, and on the number of paths of length at most 2 between *any pair of nodes*. The NP-hardness of the considered problems follows from the fact that these problems include the one considered in [18] as a special case, although we do not provide the detailed NP-hardness proofs in each specific case, since the main focus of this chapter is on analytical solutions of the considered problems rather than on a detailed complexity analysis.

4.2 Optimization Problem Formulations

Suppose that given a graph $G = (V, E)$ and a set of edges E_c on V ($E_c \cap E = \emptyset$) our goal is to augment this graph with some additional edges from E_c, so that the new network will be a 2-club/R-core, or an R-robust 2-club depending on a desired type of attack tolerance. Assume that an addition of a new edge (i, j) between nodes i and j is associated with a fixed cost c_{ij} and $G = (V, E \cup E_c)$ must be a 2-club/R-core or an R-robust 2-club. The objective is to minimize the total cost of such a network augmentation. Note that if the adjacency matrix A of $G = (V, E)$ ($A = \left(a_{ij}\right)_{i,j=1,\ldots,n}$, where $a_{ij} = 1$ if edge (i, j) initially exists and $a_{ij} = 0$ otherwise) is a null matrix, then we have a problem of optimal network design.

Let x_{ij}, $i, j = 1, \ldots, n$ be the binary variables representing the decision if an edge (i, j) is constructed. A matrix X ($X = (x_{ij})_{i,j=1,\ldots,n}$) of optimal values of the

defined variables represents an adjacency matrix of the desired network. In order for the constructed network to be 2-club/R-core, two sets of constraints need to be satisfied. The first one is that every pair of nodes i and j has to be connected directly, or through an intermediary node. It can be written as

$$x_{ij} + \sum_{k=1}^{n} x_{ik} x_{kj} \geq 1$$

for every pair (i, j), thus, we have $n(n-1)/2$ constraints in the first set. The second one ensures that every node i has at least degree R. It can be written as

$$\sum_{i=1}^{n} x_{ij} \geq R$$

for every node j, thus, we have n constraints in the second set. The optimization problem formulation is as follows.

Problem A (optimal 2-club/R-core network design/augmentation)

$$\min_{(x_{ij}: i,j=1,\ldots,n)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij} x_{ij}$$

subject to

$$x_{ij} + \sum_{k=1}^{n} x_{ik} x_{kj} \geq 1, \sum_{i=1}^{n} x_{ij} \geq R, \quad i < j = 1, \ldots, n,$$
$$x_{ij} = a_{ij}, \quad \forall (i, j) \notin E_c,$$
$$x_{ij} \in \{0, 1\}, \quad i, j = 1, \ldots, n.$$

The problem of optimal R-robust 2-club network design/augmentation can be easily formulated as follows:

Problem B (optimal R-robust 2-club network design/augmentation)

$$\min_{(x_{ij}: i,j=1,\ldots,n)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij} x_{ij}$$

subject to

$$x_{ij} + \sum_{k=1}^{n} x_{ik} x_{kj} \geq R, \quad i < j = 1, \ldots, n,$$
$$x_{ij} = a_{ij}, \quad \forall (i, j) \notin E_c,$$
$$x_{ij} \in \{0, 1\}, \quad i, j = 1, \ldots, n.$$

The R-robust 2-club and the regular 2-club constraints in both problems are quadratic. The simplest linearization method leads us to the following linearized problem:

Problem B (linearized)

$$\min_{(w_{ijk},x_{ij})} \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij} x_{ij}$$

subject to

$$x_{ij} + \sum_{k\neq i,j;k=1}^{n} w_{ikj} \geq R, \quad i < j = 1,\ldots,n,$$

$$w_{ikj} \leq x_{ik}, w_{ikj} \leq x_{kj}, w_{ikj} \geq x_{ik} + x_{kj} - 1, \quad i,j,k = 1,\ldots,n.$$

$$x_{ij} = a_{ij}, \quad \forall (i,j) \notin E_c,$$

$$w_{ikj}, x_{ij} \in \{0,1\}, \quad i,j,k = 1,\ldots,n.$$

The linearization of Problem A is the same.

This simple linearization technique of the diameter constraints works well for the considered problems with $k = 2$; however, it may be appropriate to use more advanced linearization techniques for related problems in the general case with $k > 2$. For instance, an efficient linearization technique for multi-quadratic 0–1 problems has been proposed in [10]. A compact linearization procedure for the related *maximum k-club problem* ($k > 2$), which finds the maximum k-club in a given network, has been proposed in [25].

4.3 Illustrative Examples

As mentioned above, the considered optimization problems are NP-hard, and it turns out that they are computationally challenging even for moderate size networks. Since the focus of this chapter is on theoretical foundations for low-diameter attack-tolerant networks, rather than on computational algorithms, we have conducted illustrative computational experiments for relatively small values of n to demonstrate the solution patterns of the considered network design and augmentation problems. For illustrative purposes (that will also be clarified and revisited from an analytical perspective in the next section) we assumed that each edge has the same construction cost.

In the first example, we assumed that the required attack-tolerant 2-hop network needs to be designed "from scratch," in other words, the initial graph $G = (V, E)$ has an empty set of edges. CPLEX was used to find an optimal 2-club/2-core in a graph with an even and odd number of nodes n ($n = 10, 11$) using the

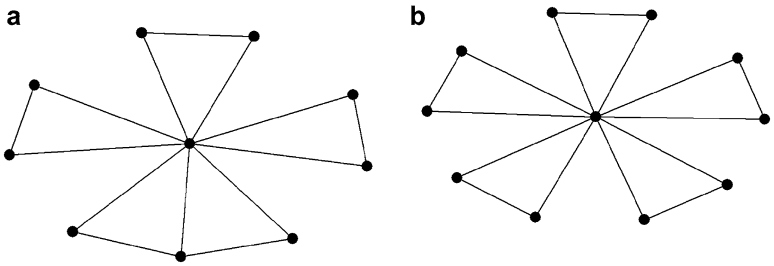

Fig. 4 Optimal 2-club/2-core network design for (**a**) $n = 10$, and (**b**) $n = 11$ nodes

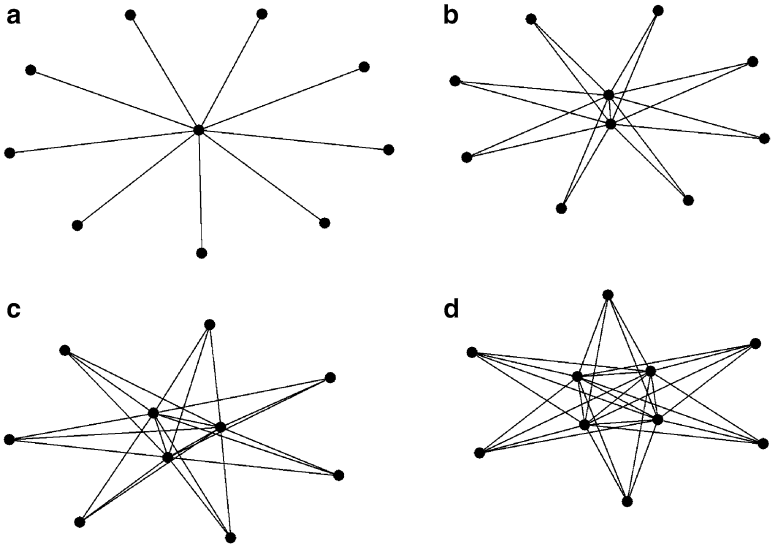

Fig. 5 Optimal 2-club (**a**), 2-robust 2-club (**b**), 3-robust 2-club (**c**), and 4-robust 2-club (**d**) network design for $n = 10$ nodes

linearized formulation of Problem A from the previous section. Further, optimal R-robust 2-clubs were identified using the formulation of Problem B for $n = 10$ and $R = 1, 2, 3, 4$.

Figures 4a and 4b show the solutions for optimal 2-club/2-core design for $n = 10$, and $n = 11$. These solutions have a clear pattern, which can roughly be described as follows: one node (hub) connects to any other nodes directly, and other nodes form pairs connected with each other and with the hub node.

Figures 5a–d show the solutions for optimal R-robust 2-club design for $R = 1, 2, 3, 4$, and $n = 10$. These solutions also have easily interpretable patterns. In the optimal R-robust 2-club there are R nodes ("hubs") directly connected to any other nodes, so R nodes in this network have a degree of $n - 1$ and $n - R$ other nodes have a degree of R.

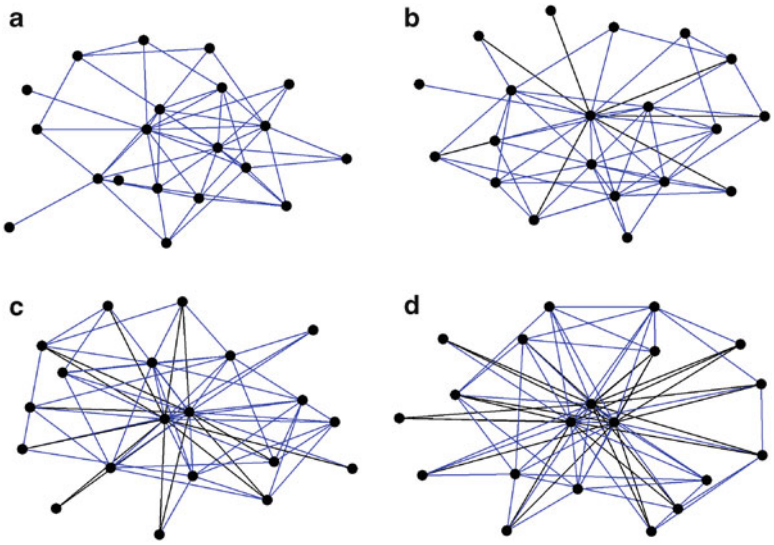

Fig. 6 Power-law network with $\beta = 0.5$ (**a**), optimal 2-club (**b**), 2-robust 2-club (**c**), and 3-robust 2-club (**d**) network augmentation for $n = 20$ nodes

In the next section, the patterns observed in Figs. 4 and 5 will be further analyzed in terms of their theoretical optimality.

Figures 6b, 6c, and 6d show the solutions for optimal R-robust 2-club *augmentation* of an existing network ($R = 1, 2, 3$). For illustrative purposes, we randomly generated a small power-law[1] network with $\beta = 0.5$ and $n = 20$. An initial network (Fig. 6a) has 53 edges. Then we solved proposed optimization problems for optimal R-robust 2-club augmentation of an existing network for $R = 1, 2, 3$. To augment this network to be a 2-club adding the minimum number of edges, one needs to add only 7 edges (Fig. 6b, the added edges are in black). For an optimal 2-robust 2-club augmentation the minimum number of edges is 15 (Fig. 6c), so by adding just 8 more edges (compared to the 2-club augmentation we are able not only to ensure that the network has a small diameter, but also to guarantee that this network will have a "strong" attack tolerance property. For an optimal 3-robust 2-club augmentation, the minimum number of edges is 25 (Fig. 6d). Note that the required number of new edges is rather small compared to the original number of edges in this network; moreover, the original network not only does not have any attack tolerance properties, but it is not even a 2-club.

[1] A power-law network with a parameter β is a network where the number of nodes with a degree k is proportional to $k^{-\beta}$.

5 Sharp Lower Bounds on the Number of Edges in 2-club/2-core and R-robust 2-club

This section presents the results on theoretical bounds and analytical optimal solutions for the problems of optimal design of weakly and strongly attack-tolerant 2-clubs. The following proposition establishes the lower bound on the number of edges in any 2-club/2-core. The ensuing corollary presents 2-club/2-cores with a clear pattern requiring *exactly* $\lceil (3n-3)/2 \rceil$ edges.

Proposition 3. *Let $G = (V, E)$ be a 2-club/2-core with $n > 6$ ($n = |V|$), then $|E| \geq (3n-3)/2$.*

Proof. Let $deg(j)$ be a degree of node j in the graph $G = (V, E)$. Pick a node i so that

$$\forall j \in V, deg(j) \geq deg(i).$$

In other words, node i has a minimum degree in $G = (V, E)$. If $deg(i) \geq 3$, then $|E| \geq 3n/2 > (3n-3)/2$ and the proposition is valid. Therefore, we only need to consider the case where $deg(i) = 2$, since in a 2-core any node has a degree of least 2.

Let $N(i)$ be the neighborhood of node i, formally defined as

$$N(i) = \{j \in V : (i, j) \in E\}.$$

Since we know that $deg(i) = 2$, then let $N(i) = \{s, t\}$. Define $S = N(s) \setminus \{i, s, t\}$, and $T = N(t) \setminus \{i, s, t\}$. Note that since $G = (V, E)$ is a 2-club, then $V = \{i, s, t\} \cup T \cup S$, and $|T \cup S| = n - 3$. Without loss of generality we always assume that $|T| \geq |S|$.

Consider 2 different cases:

- $T \cap S \neq S$, and $|T \cap S| = k$, $|S| > 0$;
- $T \cap S = S$, and $|S| = k$.

In these two cases k might be equal to zero, indicating that either sets T and S have no common nodes, or the set S is empty.

In the first case let $T' = T \setminus S$, and $S' = S \setminus T$. Both sets T' and S' are nonempty. By the definitions of T, S as "neighbors" of nodes t, s there are at least $n - 3 - k$ ($|T' \cup S'| = n - 3 - k$) edges going from $\{t, s\}$ to $T' \cup S'$. Plus there are $2k$ edges going from $\{t, s\}$ to $T \cap S$. Totally, we have $n - 3 + k$ edges going from to $\{t, s\}$ to $T \cup S$. Also, we know that any node in T' should be connected to any node in S' directly, or through some intermediary node. This intermediary node can only belong to $T \cup S$, so any path is contained in $T \cup S$. Thus, from any node $k \in T'$ there is a path in $T \cup S$ to any other node in S', and from any node $k \in S'$ there is a path in $T \cup S$ to any other node in T'. Therefore, there are at least $|T' \cup S'| - 1 = n - 4 - k$ edges in the subgraph $T \cup S$. Then, totally, we have at least $2 + (n-3+k) + (n-4-k) = 2n-5$ edges in $G = (V, E)$, which is greater than $(3n-3)/2$ for $n > 6$ (Fig. 7).

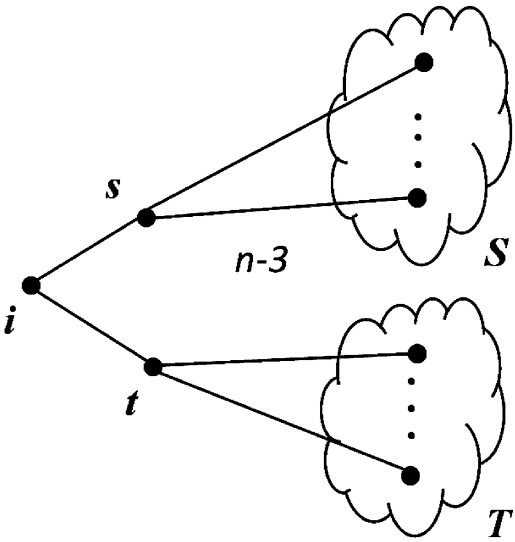

Fig. 7 Illustration of the nonoverlapping sets of edges and nodes used in the proof of Proposition 3

In the second case we have a set S as a subset of T. Let also $T' = T \setminus S$ ($|T'| = n - 3 - k$) and fix $n - 3 - k$ edges going from node t to any node in T', and $2k$ edges going from t, s to S. By definition of 2-club, there should be a path from node s to any node T'. If there is no edge (s, t), then all these paths are going through S, what requires $k(n - k - 3)$ edges going from S to T'. Then, totally, $|E| \geq 2 + (n - 3 - k) + 2k + k(n - k - 3)$ what is greater than $(3n - 3)/2$ for any $k > 0$ (with k=0 the edge (s, t) must exist).

Now assume that the edge (s, t) exists in this graph and recall that $G = (V, E)$ is a 2-core, so every node in T' should have a degree of at least 2. But we have only counted a degree of 1 of each node in T'. Therefore, there should be at least $(n-3-k)/2$ edges in T'. Thus, totally, we have $3+(n-3-k)+2k+(n-3-k)/2$ edges, what is equal to $(3n - 3 + k)/2$.

So, $G = (V, E)$ should have at least $(3n-3)/2$ edges. This ends the proof of the proposition. □

Corollary 1. *Let $G = (V, E)$ be a graph with the degree sequence vector $w = (w_1, \ldots, w_n)$, where*

$$w_i = \begin{cases} n - 1, i = 1, \\ 2, i = 2, \ldots, n - 1, \end{cases} \qquad w_n = \begin{cases} 2, \textit{if } n \textit{ is odd}, \\ 3, \textit{if } n \textit{ is even}, \end{cases}$$

and $n > 6$, then $G = (V, E)$ is the optimal 2-club/2-core in terms of the number of edges.

These results prove that the optimal patterns for 2-club/2-cores obtained in the illustrative examples (Fig. 4) in the previous section hold for any number of nodes n.

The optimal 2-club/2-core structure can be viewed as an extension of an optimal 2-club (hub-and-spoke structure depicted in Fig. 5a). To upgrade an optimal 2-club to an optimal 2-club/2-core, one needs to add $\left\lceil \frac{n-1}{2} \right\rceil$ extra edges to connect pairs of non-hub nodes and ensure that this network has a weak edge attack tolerance property. Note that an optimal 2-club/2-core has only a weak *edge* attack tolerance property, which means that it just stays connected if any one edge is destroyed. The observed pattern lacks a node attack tolerance property, i.e., the network will be disconnected if the hub node is destroyed.

The next proposition establishes the lower bound on the number of edges required to construct a *strongly attack-tolerant* 2-hop network (an R-robust 2-club). The ensuing corollary shows that the obtained lower bound is sharp. Note that this problem (in slightly different terminology) was independently considered in [21].

Proposition 4. *Let $G = (V, E)$ be an R-robust 2-club with $n \geq \frac{3R+\sqrt{5R^2-4R}}{2}$ ($n = |V|$), then $|E| \geq Rn - (R(R+1)/2)$.*

Proof. First, note that in any R-robust 2-club a degree of any node is at least R. Let $deg(j)$ be a degree of node j in the graph $G = (V, E)$. Pick a node i so that

$$\forall j \in V, deg(j) \geq deg(i)\,.$$

In other words, node i has a minimum degree in $G = (V, E)$, say $R + k$ ($k \geq 0$). Let $N(i)$ be a neighborhood of node i, i.e.,

$$N(i) = \{j \in V : (i, j) \in E\}\,.$$

Let us divide all nodes V into three nonoverlapping subsets: i, $N(i)$ and $V \setminus (N(i) \cup i)$. Note that $|N(i)| = R + k$, and $|V \setminus (N(i) \cup i)| = n - R - k - 1$. We use this division to find some nonoverlapping subsets of edges in E to establish a lower bound on $|E|$. These subsets of edges are described below item by item with the bounds on the number of edges they have.

- There are $R + k$ edges between subgraphs i and $N(i)$.
- By definition of R-robust 2-club, for any node $j \in N(i)$ there are at least R distinct paths between nodes i and j of length at most 2. Obviously, all these paths belong to subgraph $i \cup N(i)$; then, the degree of node j in the subgraph $i \cup N(i)$ is at least R. Since there is an edge $(i, j) \in E$, then the degree of node j in the subgraph $N(i)$ is at least $R - 1$ Thus, the subgraph $N(i)$ has at least $(R + k)(R - 1)/2$ edges.
- By definition of R-robust 2-club, for any node $j \in V \setminus (N(i) \cup i)$ there are at least R distinct paths between nodes i and j of length at most 2. Since there is no edge between nodes i and j ($(i, j) \notin E$), then all the paths from node i to j of length at most 2 go through $N(i)$. Hence, the number of edges between subgraphs j and $N(i)$ is at least R. Thus, there are at least $R(n - R - k - 1)$ edges between subgraphs $N(i)$ and $V \setminus (N(i) \cup i)$. Assume that we fix exactly R number of edges going from j to $N(i)$, or $R(n - R - k - 1)$ edges between subgraphs $N(i)$ and $V \setminus (N(i) \cup i)$.

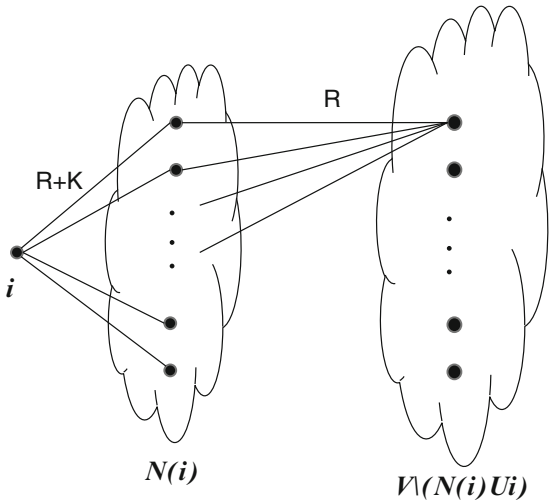

Fig. 8 Illustration of the nonoverlapping sets of edges and nodes used in the proof of Proposition 4

- A degree of any node $j \in V \setminus (N(i) \cup i)$ is at least $R + k$. We have already fixed R number of edges going from j to $N(i)$. Therefore, other k edges within the node j can either be within subgraph $N(i)$, or $V \setminus (N(i) \cup i)$. Thus, there should be at least $k(n - R - k - 1)/2$ edges that we did not count by this time.

The subsets of edges in all four items that are described above are nonoverlapping (Fig. 8). Hence,

$$|E| \geq R + k + \frac{(R + k)(R - 1)}{2} + (n - R - k - 1)R + \frac{k(n - R - k - 1)}{2},$$

which after doing arithmetic operations becomes

$$|E| \geq Rn - R(R + 1)/2 + k(n - 2R - k)/2. \tag{1}$$

If $n \geq 2R + k$, then the last part in the right-hand side of inequality (1) is nonnegative; thus,

$$|E| \geq Rn - R(R + 1)/2, \tag{2}$$

and the proposition is proven.

If $n \leq 2R + k$, then we consider another bound on $|E|$. Note that $R + k$ is the minimum degree of any node in the graph $G(V, E)$. Then, the total number of edges in this graph is bounded by

$$|E| \geq \frac{(R + k)n}{2}. \tag{3}$$

Since $n \leq 2R + k$, then $R + k \geq n - R$. Therefore, inequality (3) can be rewritten as

$$|E| \geq \frac{(n - R)n}{2}. \tag{4}$$

To see how that relates to (2) we can rewrite it as

$$|E| \geq Rn - R(R+1)/2 + \frac{n^2 - 3Rn + R^2 + R}{2}. \tag{5}$$

Using the formula for the root of the quadratic equation, we can conclude that with

$$n \geq \frac{3R + \sqrt{5R^2 - 4R}}{2}$$

the last term in the right-hand side of inequality (5) is nonnegative; thus, (2) holds, and the proposition has been proven. □

Corollary 2. *Let $G = (V, E)$ be a graph with the degree sequence vector $w = (w_1, \ldots, w_n)$, where*

$$w_i = \begin{cases} R, i = 1, \ldots, n - R \\ n - 1, i = n - R + 1, \ldots, n, \end{cases}$$

and $n \geq \frac{3R+\sqrt{5R^2-4R}}{2}$, then $G = (V, E)$ is the optimal R-robust 2-club in terms of the number of edges.

These results prove that the patterns for optimal R-robust 2-clubs observed in the previous section were also not a coincidence, and the optimal solutions have this structure for any n and require exactly $nR - \frac{R(R+1)}{2}$ edges, assuming that all edges have equal construction costs. Although it has been proven that it is the case when $n \geq \frac{3R+\sqrt{5R^2-4R}}{2}$, this condition does not seem to be very restrictive, it holds when $n \geq 3R$, and is usually satisfied for large-scale networks ($n \gg R$). Note that R-robust 2-clubs have both edge and node strong attack tolerance of level $R - 1$, which can also be observed from the above illustrations.

Note that a simple *"upgrade" procedure* can also be utilized for these network configurations. To upgrade an optimal $(R-1)$-robust 2-club to an optimal R-robust 2-club, one needs to add only $n - R - 1$ edges, effectively transforming one non-hub node into a hub node.

A final remark that immediately follows from these results is the fact that an optimal R-robust k-club (for $k > 2$) on n nodes can be constructed using *at most* $Rn - \frac{R(R+1)}{2}$ ($O(Rn)$) edges. Although the exact optimal number of edges in these structures for each specific $k > 2$ may not necessarily be derived analytically (although these issues are worth investigating in more detail), the results presented in this chapter show that R-robust k-club network configurations can *simultaneously* provide low diameter, strong attack tolerance properties with respect to both node and edge attacks and construction cost efficiency.

6 Conclusion

In this chapter, we have considered "weak" and "strong" attack tolerance properties of networks of diameter 2 (2-clubs). Maintaining both overall connectivity and small diameter after attacks on multiple network components has been identified as an important network characteristic in the situations where *all* nodes in a network need to communicate either directly or through at most one intermediary. We have shown not only that an R-robust 2-club is the necessary and sufficient structure that provides the strong edge attack tolerance property, but also that an optimal R-robust 2-club can be identified analytically in certain special cases, and that the optimal R-robust k-club configuration is cost-efficient for any $k \geq 2$ (at most $O(Rn)$ edges). We have also proved necessary and sufficient conditions for weak attack tolerance properties of diameter-2 networks and identified optimal network configurations (2-club/2-core and R-robust 2-clubs) for this type of network design problems.

Although the considered network design and augmentation problems are NP-hard (with the exception of the case with equal edge construction costs), the aforementioned analytical solutions can potentially be used to develop efficient heuristics for these problems. Overall, the focus of this chapter is on theoretical justifications of attack tolerance for low-diameter networks and on derivation of sharp lower bounds on the number of edges in these structures; however, future research should also address computational efficiency issues for the introduced problems, as well as possible extensions of the proposed concepts. In particular, analytical construction of provably optimal R-robust k-clubs for any fixed diameter $k > 2$ is a subject of special interest for future research.

Acknowledgements V. Boginski's research is partially supported by DTRA and AFOSR grants.

References

1. Abello, J., Resende, M.G.C., Sudarsky, S.: Massive quasi-clique detection. In: Lecture Notes in Computer Science, LATIN 2002: Theoretical Informatics, pp. 598–612. Springer (2002)
2. Albert, R., Jeong, H., Barabási, A.-L.: Error and attack tolerance of complex networks. Nature **406**, 378–382 (2000)
3. Balasundaram, B., Butenko, S., Trukhanov, S.: Novel approaches for analyzing biological networks. J. Comb. Optim. **10**, 23–39 (2005)
4. Balasundaram, B., Butenko, S., Hicks, I.V : Clique relaxations in social network analysis: The maximum k-plex problem. Oper. Res., **59**, 133–142 (2011)
5. Bang-Jensen, J., Chiarandini, M., Morling, P.: A computational investigation of heuristic algorithms for 2-edge-connectivity augmentation. Networks **55**(4), 299–325 (2010)
6. Bendali, F., Diarrassouba, I., Mahjoub, A.R., Mailfert, J.: The k edge-disjoint 3-hop-constrained paths polytope. Discrete Optim. **7**, 222–233 (2010)
7. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Du, D.-Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization, pp. 1–74. Kluwer Academic Publishers, Dordrecht (1999)

8. Botton, Q., Fortz, B., Gouveia, L., Poss, M.: Benders decomposition for the hop-constrainted survivable network design problem. INFORMS Journal on Computing (2011). DOI 10.1287/ijoc.1110.0472
9. Cai, G.R., Sun, Y.G.: The minimum augmentation of any graph to a k-edge-connected graph. Networks **19**, 151–172 (1989)
10. Chaovalitwongse, W., Pardalos, P.M., Prokopyev, O.A.: A new linearization technique for multi-quadratic 0–1 programming problems. Oper. Res. Lett. **32**, 517–522 (2004)
11. Dahl, G., Huygens, D., Mahjoub, A.R., Pesneau, P.: On the k edge-disjoint 2-hop-constrained paths polytope. Oper. Res. Lett. **34**, 577–582 (2006)
12. Dahl, G., Johannessen, B.: The 2-path network problem. Networks **43**, 190–199 (2004)
13. Eswaran, K., Tarjan, R.: Augmentation problems. SIAM J. Comput. **5**, 653–665 (1976)
14. Frank, A.: Augmenting graphs to meet edge connectivity requirements. SIAM J. Discrete Math. **5**, 25–53 (1992)
15. Huygens, D., Mahjoub, A.R., Pesneau, P.: Two edge-disjoint hop-constrained paths and polyhedra. SIAM J. Discrete Math. **18**(2), 287–312 (2004)
16. Hsu,T.-S.: Simpler and faster biconnectivity augmentation. J. Algorithms **45**(1), 55–71 (2002)
17. Ljubic, I.: A branch-and-cut-and-price algorithm for vertex-biconnectivity augmentation. Networks **56**(3), 169–182 (2010)
18. Li, C.-L., McCormick, S.T., Simchi-Levi, D.: On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems. Oper. Res. Lett. **11**, 303–308 (1992)
19. Luce, R.D.: Connectivity and generalized cliques in sociometric group structure. Psychometrika **15**, 169–190 (1950)
20. Mokken, R.J.: Cliques, clubs and clans. Qual. Quant. **13**, 161–173 (1979)
21. Murty, U.S.R.: On critical graphs of diameter 2. Math. Mag. **41**, 138–140 (1968)
22. Seidman, S.B., Foster, B.L.: A graph theoretic generalization of the clique concept. J. Math. Sociol. **6**, 139–154 (1978)
23. Schumacher, U.: An algorithm for construction of a k-connected graph with minimum number of edges and quasiminimal diameter. Networks **14**, 63–74 (1984)
24. Soneoka, T., Nakada, H., Imase, M.: Design of a d-connected digraph with a minimum number of edges and a quasiminimal diameter. Discrete Appl. Math. **27**, 255–265 (1990)
25. Veremyev, A., Boginski, V.: Identifying large robust network clusters via new compact formulations of maximum k-club problems. Eur. J. Oper. Res. **218**, 316–326 (2012)

Dynamics of Climate Networks

Laura C. Carpi, Patricia M. Saco, Osvaldo A. Rosso, and Martín Gómez Ravetti

Abstract A methodology to analyze dynamical changes in dynamic climate systems based on complex networks and Information Theory quantifiers is discussed. In particular, the square root of the Jensen–Shannon divergence, a measure of dissimilarity between two probability distributions, is used to quantify states in the network evolution process by means of their degree distribution. We explore the evolution of the surface air temperature (SAT) climate network on the Tropical Pacific region. We find that the proposed quantifier is able not only to capture changes in the dynamics of the studied process but also to quantify and compare states in its evolution. The dynamic network topology is investigated for temporal windows of one-year duration over the 1948–2009 period. The use of this novel methodology, allows us to consistently compare the evolving networks topologies

L.C. Carpi (✉)
Civil, Surveying and Environmental Engineering,
The University of Newcastle, Callaghan, NSW, Australia

Departamento de Física, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil
e-mail: laura@studentmail.newcastle.edu.au

P.M. Saco
Civil, Surveying and Environmental Engineering,
The University of Newcastle, Callaghan, NSW, Australia
patricia.saco@newcastle.edu.au

O.A. Rosso
Chaos & Biology Group, Instituto de Cálculo,
Universidad de Buenos Aires, Buenos Aires, Argentina

Departamento de Física, Universidade Federal de Minas Gerais,
Belo Horizonte, MG, Brazil
e-mail: oarosso@gmail.com

M.G. Ravetti
Departamento de Engenharia de Produção,
Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil
e-mail: martin.ravetti@dep.ufmg.br

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*,
Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_8,

and to capture a cyclic behavior consistent with that of El Niño/Southern Oscillation. This cyclic behavior involves alternating states of less/more efficient information transfer during El Niño/La Niña years, respectively, reflecting a higher climatic stability for La Niña years which is in agreement with current observations. The study also detects a change in the dynamics of the network structure, which coincides with the 76/77 climate shift, after which, conditions of less-efficient information transfer are more frequent and intense.

Keywords Climate networks • Complex network evolution • Information theory quantifiers • Jensen–Shannon divergence • Shannon entropy • El Niño/Southern oscillation

1 Introduction

Recent methodologies based on both complex networks and information theory have shown potential for the analysis of nonlinear processes in large data sets and, in particular, for the investigation of the collective behavior of several interrelated variables. For the case of climatic data, recent research during the last decade using complex network theory has contributed to a better understanding of the interactions of the multiple processes within the climate system. In particular, properties of well-known network models have been used to extrapolate behaviors in the climate system. For example, the presence of super-nodes was associated with teleconnection patterns [31, 33], and small-world properties were associated with a high efficiency in the transmission of information, and also with the stability of the climate system [30]. This chapter is organized as follows: Section 2 presents an introduction on climate networks. Section 3 provides a literature review on recent research and developments on climate networks. Section 4 presents the methodology and computational experiments. Section 5 presents the application of the proposed methodology to the Tropical Pacific region to evaluate whether or not it is capable to capture, in terms of information organization, the ENSO process. Finally, in Sect. 6 final remarks and conclusions are discussed.

2 Climate Networks

The study of complex networks has its origin in a branch of discrete mathematics known as graph theory. The pioneer work of the Swiss mathematician Leonhard Euler in 1736, known as the Konigsberg bridge problem, marks the beginning of the graph theory. Important developments in this field have brought solutions to many practical questions and provided the mathematical bases to the study of networks [6]. Climate networks are systems whose structure is irregular and

dynamically evolving in time. The study of these kinds of networks emerged in the last decade and has showed an increasing interest in the scientific community, with important applications in many different areas [2, 6, 21, 36]. Two seminal papers, one by Watts and Strogatz on small-world networks that appeared in Nature in 1998 [36] and another by Barabási and Albert on scale-free networks that appeared in Science in 1999 [4], the increased computing power, and the availability of data sets of observational data made possible the study of the properties of real networks. Concepts and measures to characterize different topologies of real networks were necessary for better understanding different behaviors. In this sense, the most important result has been the identification of a series of unifying principles and statistical properties that are common to most of the real networks [6]. Applications of complex network theory for the analysis of climate dynamics are very recent and started with the pioneering work of Tsonis [30]. Since then, only a few studies have been reported in the literature [11, 12, 16, 26, 27, 31–33, 35, 38].

Climate networks are systems composed by a very large (thousands or millions) number of nodes, characterized by a complex interaction and topology. This complexity is determined by the dynamics of the nodes and the characteristic of the interactions, which can be linear or nonlinear. In the same way the resulting topology can be random or with some kind of organization and also can be constant or changing in time [13].

The basis behind the construction of climate networks is the concept of synchronization from dynamical system theory that appears as the transfer of dynamical information in a complex network topology [3]. Dynamical correlations can be thought as (partial) synchronization of nonlinear oscillators on a grid spatially connected network [12]. Each point of that grid is considered to represent a dynamical system interacting in some complex way with the others, forming a network. In this case, synchronization translates into links between the individual oscillators that define the structure of the network [30]. The collective behavior can be studied through the characteristics of the resulting network structure [33]. Dynamical correlations between the nodes are usually computed using linear measures like the parametric Pearson cross-correlation coefficient or the nonparametric Spearman cross-correlation coefficient. Both coefficients measure the strength of a linear relationship between two time series, the only difference is that Spearman does not assume normality as Pearson does. Although these metrics are very efficient and widely used, they do not capture nonlinear relationships. Another limitation is the computation of interdependency between a large number of elements. To solve these two limitations, information theory quantifiers such as mutual information, which are able to capture nonlinear relationships, and complex networks measures that allow the study of a large number of elements reflecting their collective behavior can be used instead.

The first step for the construction of the network topology is the identification of an appropriate data set. Recently used data sets for the study of climate networks have been sea surface temperature (SST) [32], sea air temperature (SAT) [11, 12]; pressure, relative humidity, and precipitable water [26] and the 500 hPa data set

corresponding to the height of the 500 mb pressure level [30,33]. All these data sets are available with a good resolution (grid size) through USA National Center for Environmental Prediction NCEP/NCAR.

The second step is the computation of the degree of interdependency between all the nodes of the network for the construction of the distance matrix. Each element of the matrix represents the correlation value between two nodes. The third step is the selection of the threshold τ. The choice of an appropriated threshold is an important point in the construction of the network topology. The value of the threshold must be selected considering that statistically significant connections have to be maintained. It is important to know that different features can be revealed at different thresholds. For example, in [12] it is shown that the frequency of having strong Pearson correlation coefficients is bigger for short distance edges and this frequency reduces gradually as the distances of the edges become higher. This implies for example in the case of long distance edges or teleconnections with high correlation values will only be included in the climate network if the threshold is not too high. If the choice of the threshold is not well evaluated for a specific data set, important features of the climate network could be excluded and only short distance edges remain. Considering this, the selection of the threshold has to be a balance between the statistical importance of the connections and the features in the network that are important to be included. A deeper analysis of this issue can be found in [26]. Having the threshold selected, it is possible to construct the network topology considering the connections that correspond to pairs of nodes that satisfy this threshold.

3 Literature Review on Climate Networks

In their pioneer work, Tsonis and Roebber [30] introduce concepts of complex network theory to study patterns in the climate system. Since then, there has been an increasing interest in the scientific community in the exploration of climate networks. However, as this field is still very young, not many works have appeared in the literature. In the following some of these works are presented.

In [30] the authors consider the global National Center for Environmental Prediction/National Center for Atmospheric Research (NCEP/NCAR) reanalysis 500 hPa data set that indicates the height of the 500 mb pressure level and can be considered as a representation of the general circulation of the atmosphere. Each grid point is a node in the network and it is considered to be a dynamical system that varies in time in a complex way.

The connections between the nodes are estimated by computing the Pearson correlation coefficient at lag zero between the time series of all possible pairs of nodes. A pair is considered as connected if their correlation coefficient is above a certain threshold, defining the final structure of the network. Studying the characteristic of the network the authors found that the climate network has properties of small-world networks [36] divided into two interconnected sub-networks one of them operating

in the tropics and one operating in the high latitudes, both connected by an equatorial one that links the two hemispheres. The tropical one is an almost fully connected network, the one in the mid-latitude a scale-free network with dominant super nodes. The authors interpret that this particular network structure makes the climate system to be stable and efficient in transferring information. The small-world architecture allows the system to respond quickly to any fluctuation introduced in the system reducing the possibility of prolonged local extremes but making local events to have global implications. Tsonis et al. [33] analyze the first connection between the emergence of super nodes in the climate network and teleconnection patterns. They also discuss that the use of linear approaches like linear correlation coefficient may exclude the detection of nonlinear behaviors. The use of measures based on information theory like for example mutual information that could be used instead, requires long data sets that are not always available. As the linear correlation coefficient was used as a construction tool, it was capable to show scale-free topology properties (observed by the degree distribution) that are associated with nonlinear dynamics.

Tsonis et al. [31] further analyze the relation between super-nodes and major teleconnections patterns. They identify super-nodes located in North America and Northeast Pacific Ocean where the Pacific-North American (PNA) pattern is found. In the Southern Hemisphere the authors find super-nodes over the southern tip of South America, Antarctica, and the south Indian Ocean where the Pacific-South American (PSA) pattern is identified. The main finding in this work is that it shows that teleconnections are necessary to make the climate system stable and efficient in transferring information. The scale-free and the small-world properties of the extratropical network allow the system to respond efficiently by diffusing the information of local fluctuations, avoiding prolonged extremes, and providing stability to the system. To verify this result, they use empirical orthogonal function (EOF) analysis to construct a new data set that excludes the modes that significantly contribute to the total variability, the North Atlantic Oscillation (NAO) or Pacific North American Pattern (PNA). The authors reconstruct the climate network without these teleconnections and find that when excluding NAO the network becoming less efficient in transferring information. The authors conclude that teleconnections are key of the stability of climate system decreasing the probability of climate shifts.

In [38] different zones of the globe are considered to study whether or not they are influenced by an El Niño event. Temperatures in different zones in the world do not show significant changes due to El Niño except when measured in a restricted area in the Pacific Ocean. However, the dynamics of a climate network based on the same temperature records in four geographical zones in the world is significantly influenced by El Niño. To evaluate this, Yamasaki and coauthors construct networks from daily temperature measurements using cross-correlation coefficient between four different and distant zones of the globe. They evaluate the number of links that survive during an El Niño event. The surviving number of links give a specific and sensitive measure for El Niño events. While during non-El Niño periods these links which represent correlations between temperatures in different sites are more stable, fast fluctuations of the correlations observed during El Niño

periods cause the links to break. They found that the links that break during El Niño are mostly links that have large time delays implying that structural changes are seen even for geographical zones where the mean temperature is not affected by El Niño.

A different approach is presented in [32], where the authors compare networks constructed from surface temperature for El Niño and La Niña years to investigate differences in their structure. They separate the time series into El Niño and La Niña values if they fall into El Niño or La Niña events, and a residual time series is constructed for intermediate values. The characteristics of the networks show the presence of super-nodes, smaller in El Niño network that also has fewer number of links than La Niña network. The loss of links in El Niño network is associated with a significant decrease in the network's clustering coefficient and characteristic path length.

These properties are able to describe the network topologies indicating that El Niño is a less communicative network. The authors speculate that there is a mechanism related to global temperatures [28, 29] that makes El Niño network less communicative and less predictable.

Finally, Donges et al. [12] propose the construction of a climate network using both Pearson correlation coefficient and nonlinear mutual information from monthly averaged global surface air temperatures (SAT). They compare both networks and find a high degree of similarity in all properties with exception of the betweenness centrality, that shows more pronounced regional differences. They find a higher degree of differences in betweenness structures over the oceans, particularly over the East Pacific, the North Atlantic, and the arctic regions. They hypothesize that the betweenness centrality may quantify the local differences between networks constructed using Pearson correlation and mutual information that could find traces of nonlinear physical processes in the climate system. By definition betweenness centrality is a very sensitive measure and can locally depend heavily on the existence or nonexistence of a small number of edges in the network [1].

4 Network Analysis

In most works, the analysis of the network topology is made using complex networks elements such as: Average path length, clustering coefficient, closeness centrality, and betweenness centrality. In the following, these complex networks elements are defined as in [12].

4.1 Average Path Length

The average path length L of a graph is the average topological distance between all connected pairs of vertices. The topological distance or shortest path length d_{ij}

is the minimum number of edges that have to be crossed to travel from vertex i to vertex j. The average path length is defined as

$$L = \frac{1}{\binom{N}{2}} \sum_{i<j} d_{ij}. \tag{1}$$

4.2 Clustering Coefficient

The clustering coefficient is a measure of the tendency of nodes to cluster together and it is possible to define a local and a global measure of the clustering coefficient in a network. The local clustering coefficient is the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them [36]. It gives the probability that two randomly chosen first neighbors of i are also neighbors [12]. Considering Γ_i a set of first neighbors of i and $e(\Gamma_i)$ the number of edges connecting the vertices within the neighborhood Γ_i, the clustering coefficient can be defined as

$$C_i = \frac{e(\Gamma_i)}{\binom{k_i}{2}}, \tag{2}$$

where $\binom{k_i}{2} = \frac{1}{2}k_i(k_i - 1)$ is the maximum number of edges in Γ_i.

The global clustering coefficient C is the mean local clustering coefficient, $C = \langle C_v \rangle_v$.

4.3 Closeness Centrality

The closeness centrality CC_i is the inverse average topological distance of vertex i to all others in the network. If CC_i is large , i is topologically close to the rest of the network. Closeness centrality is normalized to $0 \leq CC_i \leq 1$, and is defined as

$$CC_i = \frac{N-1}{\sum_{i=1}^{N} d_{ij}}. \tag{3}$$

4.4 Betweenness Centrality

Betweenness is a measure of the centrality of a node in a network. It is usually calculated as the fraction of shortest paths between node pairs that pass through the node of interest. Assuming the information travels through the network in shortest

paths and there are σ_{ij} shortest paths connecting i and j, a betweenness centrality can be expressed by

$$BC_v = \sum_{i,j \neq v}^{N} \frac{\sigma_{ij(v)}}{\sigma_{ij}} \tag{4}$$

where $\sigma_{ij}(v)$ is the number of shortest paths from i to j that include v.

The above-defined elements give information about the network topology and its organization, but can they be used to compare different states when the number of links change in time? For example, how can two average path length values be compared if the networks present different number of links? With the objective of capturing the dynamic of the system, through the analysis of the network topology evolution, we use information theory quantifiers such as Shannon entropy and Jensen–Shannon divergence. For the computation of these quantifiers we use the degree distribution of the network that is a distribution function $P(k)$ which gives the probability that a randomly selected node has exactly k edges [2]. Then, $P(k)$ is defined as n_k/n, being "n" the total number of nodes in the network. These quantifiers give a global measurement of the network randomness that reflects its topological organization. They can be used especially to compare states as its computation is not affected by nodes that eventually become disconnected. These quantifiers are still poorly explored as tools to characterize complex networks. A small number of works have been reported using the concept of entropy [10, 34] and complexity [37]. We propose the use of the square root of the Jensen–Shannon divergence to characterize the evolution of networks by means of the average degree distribution. Our work is motivated by a real application, the analysis of the evolution of the El Niño/Southern Oscillation (ENSO) phenomenon. As previously discussed, coordinates in a gridded data set are considered nodes, and edges are defined by correlations between pairs of data points. This analysis can export to many different areas, in a similar fashion, a gene network will consider each gene as a node and links will be created depending on the gene-expression correlation values [7]. Those types of networks are well known to possess complex network attributes [5, 25, 30] and can be modeled with a fixed number of nodes during their evolution process (grid points or genes).

4.5 Shannon Entropy

Shannon entropy measures the degree of heterogeneity of the network [34]. Its zero value corresponds to the state of having complete knowledge of the process. In our particular case, it means that the average degree of the network is known (regular lattice). On the other hand, the maximum entropy value occurs when our knowledge of the system is minimum (uniform random network). The entropy of the degree distribution $P(k)$ can be described as $S = -\sum_k P(k) \ln P(k)$ [9].

4.6 Jensen–Shannon Divergence

Another important quantifier is the Jensen–Shannon divergence ($\mathcal{J}$). This quantifier is a measure of the dissimilarity between two probability distributions. It presents ultra-metricity properties: (1) positive values, (2) symmetry, and (3) a zero value for equal probability distributions. The only missing property to obtain a metric is the triangle inequality that can be achieved by taking its square root [14, 22]. As we are interested in quantify and compare states in a network evolution, we use $\mathcal{J}^{1/2}$.

$$\mathcal{J}[P, P_{\text{ref}}] = S[(P + P_{\text{ref}})/2] - S[P]/2 - S[P_{\text{ref}}]/2$$

As we are interested in modeling networks with a fixed number of nodes during their evolution process (grid points), the Watts–Strogatz (WS) model [36] was chosen as it is the model that fits the best to test our methodology. The WS model starts with a regular network, and during subsequent steps, each edge can be rewired to a randomly chosen vertex with a given probability p. By using this model, several intermediate states from the initial regular network (all nodes with k incident edges) to a random network are obtained. At each step of the process, the probability p is increased and the network walks towards a random graph. The probability of not rewiring a specific edge of the regular lattice is then given by $(1-p) + p(k+1)/n$, that is, the sum of the probability of not allowing that edge to change plus the probability that the chosen target node is already linked to the edge (this change is therefore prohibited). This result can be generalized for the complete network: the probability of a regular network not changing its structure in one step of the process is $P = ((1-p) + p(k+1)/n)^{nk/2}$. The binomial condition of the process should lead to a Poisson distribution, but the fact that the model considers an ongoing evolution of the network, that is, changes in the network remain for the following steps, alters the shape of the final average degree probability distribution. An alternative model, which we call herein modified WS (mWS) is also considered. The difference between the WS and mWS models is that on the latter the changes at each step are not retained, which means that every step of the process starts from the regular lattice. It is interesting noticing that the modification of the model does not alter the small-world properties of the process. However, the average degree distribution converges to a true Poisson, with parameter k for the extreme case of $p = 1$. We refer to our recent work [7] for a deeper discussion on this experiment.

For the computation of the Jensen–Shannon divergence, three different PDFs are used as a reference to compare and analyze the topological changes: a Poisson distribution (P_o with $\lambda = k$), the uniform distribution P_e, and the PDF corresponding to the regular lattice P_r. Both P_e and P_r are extreme and invariant cases, P_e corresponds to the asymptotic random network stage and P_r to the initial stage of the process. Poisson was chosen as it is the PDF reached by the Erdös–Rényi random model [15] and it has been used by many authors as possible average degree distribution. The use of the regular lattice is interesting as it is the only extreme case that can be achieved by a single real network. On the other hand, for practical purposes the uniform PDF is appealing as its values are independent of the number of edges of the network.

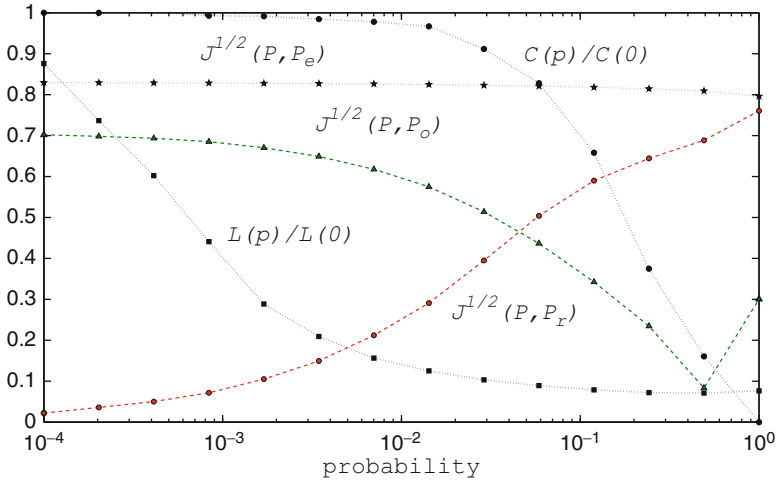

Fig. 1 Normalized characteristic path length (L(p)/L(0)), normalized clustering coefficient (C(p)/C(0)), and square root of the Jensen–Shannon divergence $\mathcal{J}^{1/2}[P, P_{\text{ref}}]$ for the WS model. The initial stage is a regular lattice of 1,000 nodes, each one with degree 10. For each value p, 50 trials were averaged to compute the degree distribution. $\mathcal{J}^{1/2}[P, P_{\text{ref}}]$ is obtained from the average degree distribution using $P_{\text{ref}} = \{P_o, P_e, P_r\}$

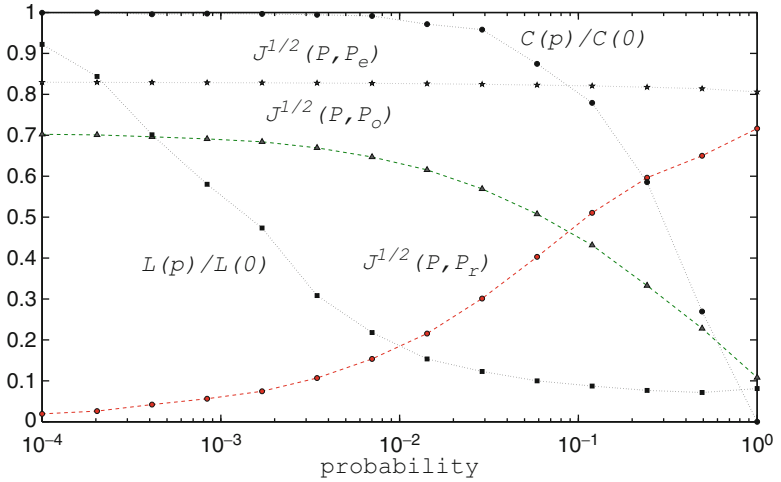

Fig. 2 Same as Fig. 1 but considering the mWS model

Figures 1 and 2 display the $\mathcal{J}^{1/2}[P, P_{\text{ref}}]$ values for the WS and the mWS models. These figures show that the average path length (L) and the clustering coefficient (C) have similar behavior for both models and their values are in agreement with those presented in [36]. As $\mathcal{J}^{1/2}[P, P_r]$ and $\mathcal{J}^{1/2}[P, P_e]$ are extreme cases in the

progression, they present unique values for each probability p independently of the model considered. The $\mathcal{J}^{1/2}[P, P_o]$ only has unique values when using the mWS model and the average PDF tends to reach a $P_o(k)$ and diverges from it, confirming the results by Newman et al. [19]; the experiment here discussed contemplates networks with 1,000 nodes and $k = 10$ as discussed in the seminal article of Watts and Strogatz [36]. We refer to [7] for a deeper discussion on this experiment including the use of statistical complexity for the analysis of the network topology evolution.

Depending on the ratio between the number of nodes and neighbors (n/k), the network may become disconnected during the process, this is also common on real-world applications. When this happens the average path length, must be adjusted [20]. In the case of the $\mathcal{J}^{1/2}$, the existence of disconnected nodes does not interfere in the network analysis. Furthermore, the metric property of the $\mathcal{J}^{1/2}$ can be used, not only as a tool to measure how far our network structure is from the chosen reference, but also to compare different states in the evolution, or to measure the distance between two different network structures. The only requirement is that they must maintain the number of nodes, for a fair comparison, as their PDF's dimensions will be the same.

5 Analysis of the ENSO Phenomenon

For this analysis, we consider the Tropical Pacific $(120E° - 70W°, 20N° - 20S°)$ monthly averaged surface air temperature (SAT) reanalysis data set [17]. We chose this data to maintain consistency with previous works [30, 38], and because it captures the dynamics on the interface between ocean and atmosphere due to heat exchange [12]. This data set is therefore appropriate to investigate the evolution of the El Niño-Southern Oscillation (ENSO). The ENSO cycle takes from 3 to 4 years (average), its warm and cold phases are called El Niño and La Niña, respectively.

The climate network analyzed here is constructed using monthly averaged surface air temperature (SAT) data over the Tropical Pacific region $(120E° - 70W°, 20N° - 20S°)$ for the period 1948–2009. The network structure constructed using SAT data has also been used in previous studies to enable capturing the dynamics of the heat exchange at interface between ocean and atmosphere [12, 32]. The data set used corresponds to the reanalysis data distributed by the National Center for Environmental Prediction/National Center for Atmospheric Research (NCEP/NCAR), which is organized on a grid with resolution of 2.5×2.5 (lat-lon) [17]. Consequently, the resulting grid for the Tropical Pacific region has a total of $1,156$ nodes $(17 \times 68$ nodes). The evolution of the network topology, from 1948 to 2009, was followed by considering 62 annual nonoverlapping windows corresponding to the January to December monthly values.

The network topology is obtained by computing Spearman correlation between pairs of points (nodes), with edges created when correlation values exceed 0.9. The evolution of the network topology is captured by considering annual sliding

windows and obtaining the degree distribution for each of them. Temporal changes are then analyzed by computing the square root of the Jensen–Shannon divergence. We refer to [8] for a complete analysis of this application including a sensibility analysis of the threshold selection.

Figure 3 shows the temporal evolution of the climate network topology as captured by the standard complex network quantifiers: number of links (a), average clustering coefficient (b), average path length (c), and efficiency (d). This figure also shows years corresponding to strong El Niño and La Niña events, identified using the Oceanic Niño Index (ONI). The ONI is a three-month running mean of the reconstructed sea surface temperature (SST) anomalies in the Niño 3.4 region [24]. Values of ONI are available through the National Oceanic and Atmospheric Administration (NOAA) climate prediction center (http://www.cpc.noaa.gov).

As seen from Fig. 3, throughout the study period, the dynamic climate network has large average clustering coefficient and small average path length values; these network properties are consistent with those of small-world networks frequently found in real-world systems [12]. This figure also shows that temporal variations in all these measures reflect a cyclic behavior consistent with that of ENSO. There is a clear tendency for networks obtained for all the strong La Niña years to display lower average clustering coefficients, higher average path lengths, and lower number of links than the networks corresponding to strong El Niño years. As expected for networks with fewer links and higher average path length, the efficiency for El Niño years is lower (Fig. 3d) than that of La Niña years.

Figure 4 shows the temporal variability of $\mathcal{J}^{1/2}$, also reflecting the ENSO cyclic behavior. In this figure, we also show years corresponding to both strong and moderate, El Niño and La Niña events, identified using ONI. As mentioned before, the metric properties of the $\mathcal{J}^{1/2}$ quantifier and its independence from the total number of links make it particularly suitable for comparing the characteristics of the changing network topology analyzed in this study, where the number of links changes with time. Though the degree distribution maintains approximately the same distance to the reference uniform distribution P_e throughout the study period, the $\mathcal{J}^{1/2}$ values corresponding to all moderate and strong La Niña and El Niño years are, respectively, below and above the average value of $\mathcal{J}^{1/2}$ (horizontal dashed line). This means that the structure for El Niño years is closer to that of regular networks, and therefore less efficient in transferring information.

It is important to note that the efficiency of the climate network can be interpreted in terms of potential effects that local fluctuations can have at global climate scales. Fluctuations tend to destabilize the source region, and these fluctuations, which are equivalent to information in network analysis, are transferred through the network. If this transfer is efficient then the possibility of prolonged local fluctuations (as for example local extremes) is reduced, providing more stability to the system [31]. Therefore, more regular structures, as those corresponding to El Niño years, could be associated to strong local events that are not efficiently dampened by the network structure.

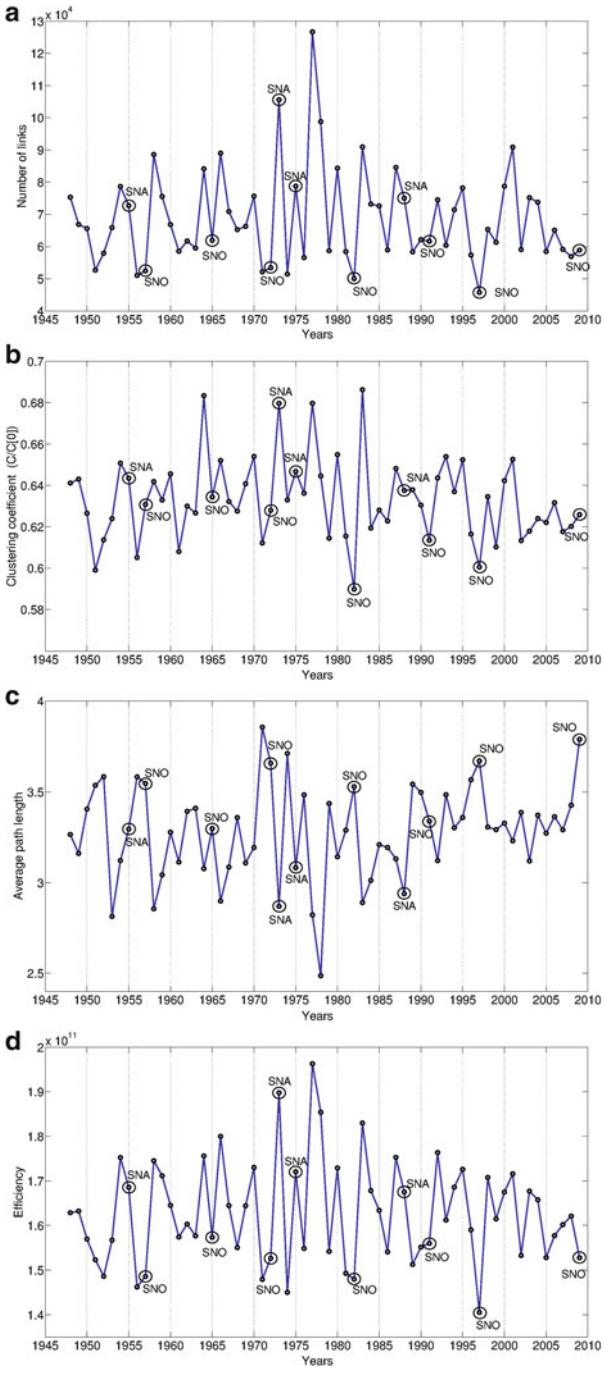

Fig. 3 Evolution of network topology as captured by: (**a**) number of links, (**b**) clustering coefficient, (**c**) average path length, and (**d**) efficiency. Strong recorded ENSO events are indicated as SNO for El Niño and SNA for La Niña

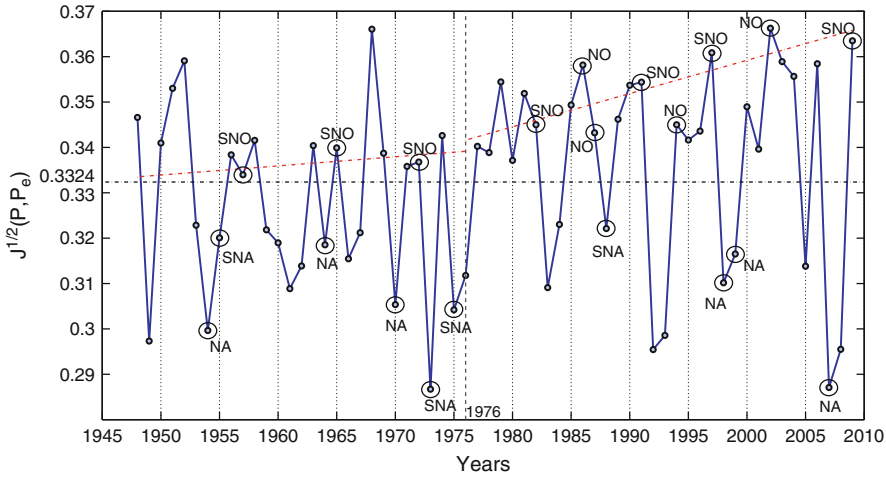

Fig. 4 Evolution of the square root of the Jensen–Shannon divergence, $\mathcal{J}^{1/2}(P, P_e)$, for the Tropical Pacific region. Strong and moderate ENSO events are indicated as SNO and NO for El Niño and SNA and NA for La Niña respectively. The *vertical dashed line* indicates the 76/77 climate shift and the *red lines* show trends in $\mathcal{J}^{1/2}(P, P_e)$ computed for all El Niño events before and after the shift

Another interesting observation, evident from the dynamical analysis of the network structure and captured by the evolution $\mathcal{J}^{1/2}$ displayed in Fig. 4, is a change in dynamics occurring approximately after 76/77. This change in the dynamics of the network structure coincides with the 76/77 climate shift extensively discussed in the literature [18, 23]. As noted in the literature, the intensity and frequency of the El Niño events increased after the climate shift. Our analysis detects that this climate shift gives rise, on average, to a more regular climate network as shown by the more frequent values of $\mathcal{J}^{1/2}$ above the horizontal line after 76/77. The red lines in Fig. 4 show the linear trends fitted to the values of $\mathcal{J}^{1/2}$ for El Niño events before and after 1976. These trends highlight that the values of $\mathcal{J}^{1/2}$ for El Niño events are not only more frequent but also higher for the post-shift period. Therefore, the network after the climate shift exhibits conditions of less efficient information transfer that can be associated to a less stable climate with local extreme events, which are more frequent and intense than those previous to the shift.

6 Final Remarks

We present in this work a novel and integrative approach that enabled us to investigate the temporal evolution of a climate network by dynamically analyzing the topologies constructed in temporal windows of one-year duration over the 1948–2009 record. This methodology based on complex network analysis and Information

Theory quantifiers showed to be able to reflect dynamical changes in network topologies characterizing the evolution of the system. The use of the $\mathcal{J}^{1/2}$ it is useful specially when nodes in the network dynamically change their connections over time, as this fact does not interfere its computation. The $\mathcal{J}^{1/2}$ value can be thought as a global characteristic of the network. As it has metric properties, it can be used, not only as a tool to measure how far the network structure is from the chosen reference, but also to compare different states during its evolution.

In the system here studied, the Tropical Pacific climate network, the $\mathcal{J}^{1/2}$ was capable to reflect the structural changes during the its evolution and to characterize El Niño and La Niña events. We found that the network topologies clearly display a cyclic behavior consistent with that of El Niño/Southern Oscillation (ENSO). The strong and moderate El Niño events studied by their networks topologies exhibit closer distances to a regular network structure than those of the strong and moderate La Niña events. The study also detects a change in the dynamics of the network structure, which coincides with the 76/77 climate shift. The network after the climate shift exhibits conditions of less efficient information transfer, which are more frequent and intense, can be associated to a less stable climate.

Acknowledgements This research has been supported by a scholarship from The University of Newcastle awarded to Laura C. Carpi. O.A. Rosso acknowledges support from CONICET, Argentina and CAPES, Brazil. And M.G. Ravetti from FAPEMIG and CNPq, Brazil.

References

1. Albert, R., Albert, I., Nakarado, G.L.: Structural vulnerability of the North American power grid. Phys. Rev. E **69**(2), 025103– (2004)
2. Albert, R., Barabasi, A.-L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47 (2002)
3. Arenas, A., Diaz-Guilera, A., Kurths, J., Moreno, Y., Zhou, C.: Synchronization in complex networks. Phys. Rep. **3**, 93 (2008)
4. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
5. Barrenas, F., Chavali, S., Holme, P., Mobini, R., Benson, M.: Network properties of complex human disease genes identified through genome-wide association studies. PLoS ONE **4**(11), e8090 (2009)
6. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.: Complex networks: Structure and dynamics. Phys. Rep. **424**(4–5), 175–308 (2006)
7. Carpi, L., Rosso, O.A., Saco, P.M., Ravetti, M.G.: Analyzing complex networks evolution through information theory quantifiers. Phys. Lett. A **375**, 801–804 (2011)
8. Carpi, L., Rosso, O.A., Saco, P.M., Ravetti, M.G.: Structural evolution of the tropical pacific climate network. Submitted for publication (2012)
9. Costa, L.d.F., Rodrigues, F.A., Travieso, G., Villas Boas, P.R.: Characterization of complex networks: A survey of measurements. Adv. Phys. **56**(1), 167–242 (2007)
10. Demetrius, L., Manke, T.: Robustness and network evolution—An entropic principle. Physica A, **346**(3–4), 682–696 (2005)

11. Donges, J.F., Zou, Y., Marwan, N., Kurths, J.: The backbone of the climate network. Europhys. Lett. **87**(4), 48007 (6pp) (2009)
12. Donges, J.F., Zou, Y., Marwan, N., Kurths, J.: Complex networks in climate dynamics. Eur. Phys. J. Spec. Top. **174**(1), 157–179 (2009)
13. Donner, R., Barbosa, S., Kurths, J., Marwan, N.: Understanding the earth as a complex system recent advances in data analysis and modelling in earth sciences. Eur. Phys. J. Spec. Top. **174**(1), 1–9 (2009)
14. Endres, D.M., Schindelin, J.E.: A new metric for probability distributions. IEEE Trans. Inform. Theory **49**(7), 1858–1860 (2003)
15. Erdös, P., Rényi, A.: On random graphs, i. Publicationes Mathematicae (Debrecen) **6**, 290–297 (1959)
16. Gozolchiani, A., Yamasaki, K., Gazit, O., Havlin, S.: Pattern of climate network blinking links follows elniño events. Europhys. Lett. **83**(2), 28005 (2008)
17. Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., Zhu, Y., Leetmaa, A., Reynolds, R., Chelliah, M., Ebisuzaki, W., Higgins, W., Janowiak, J., Mo, K.C., Ropelewski, C., Wang, J., Jenne, R., Joseph, D.: The ncep/ncar 40-year reanalysis project. Bull. Am. Meteorol. Soc. **77**(3), 437–471 (1996)
18. Lee, T., McPhaden, M.J.: Increasing intensity of el niño in the central-equatorial pacific. Geophys. Res. Lett. **37**, L14603 (2010)
19. Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Random graphs with arbitrary degree distributions and their applications. Phys. Rev. E **64**(2), 026118– (2001)
20. Newman, M.E.J., Watts, D.J.: Renormalization group analysis of the small-world network model. Phys. Lett. A **263**, 341–346 (1999)
21. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**, 167 (2003)
22. Österreicher, F., Vajda, I.: A new class of metric divergences on probability spaces and its applicability in statistics. Ann. Inst. Stat. Math. **55**(3), 639–653 (2003)
23. Ren, H.L., Jin, F.F.: Niño indices for two types of ENSO. Geophys. Res. Lett. **38**, L04704 (2011)
24. Smith, T.M., Reynolds, R.W., Peterson, T.C., Lawrimore, J.: Improvements to NOAA's historical merged land–ocean surface temperature analysis (1880–2006). J. Climate **21**(10), 2283–2296 (2008)
25. Stam, C.J., Jones, B.F., Nolte, G., Breakspear, M., Scheltens, Ph.: Small-world networks and functional connectivity in alzheimer's disease. Cereb. Cortex **17**(1), 92–99 (2007)
26. Steinhaeuser, K., Chawla, N.V., Ganguly, A.R.: An exploration of climate data using complex networks. In: SensorKDD '09: Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data, pp. 23–31. ACM, New York (2009)
27. Swanson, K.L., Tsonis, A.A.: Has the climate recently shifted? Geophys. Res. Lett. **36**, L06711 (2009)
28. Tsonis, A.A., Elsner, J.B., Hunt, A.G., Jagger, T.H.: Unfolding the relation between global temperature and ENSO. Geophys. Res. Lett. **32**(9), L09701– (2005)
29. Tsonis, A.A., Hunt, A.G., Elsner, J.B.: On the relation between ENSO and global climate change. Meteorol. Atmos. Phys. **84**(3), 229–242 (2003)
30. Tsonis, A.A., Roebber, P.J.: The architecture of the climate network. Physica A **333**, 497–504 (2004)
31. Tsonis, A.A., Swanson, K.L., Wang, G.: On the role of atmospheric teleconnections in climate. J. Climate **21**, 2990–3001 (2008)
32. Tsonis, A.A., Swanson, K.L.: Topology and predictability of el niño and la niña networks. Phys. Rev. Lett. **100**, 228502 (2008)
33. Tsonis, A.A., Swanson, K.L., Roebber, P.J.: What do networks have to do with climate? Bull. Am. Meteorol. Soc. **87**(5), 585–595 (2006)
34. Wang, B., Tang, H., Guo, C., Xiu, Z.: Entropy optimization of scale-free networks robustness to random failures. Physica A **363**, 591–596 (2006)

35. Wang, G., Swanson, K.L., Tsonis, A.A.: The pacemaker of major climate shifts. Geophys. Res. Lett. **36**, L07708 (2009)
36. Watts, D.J., Strogatz, D.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
37. Wilhelm, T., Hollunder, J.: Information theoretic description of networks. Physica A **385**(1), 385–396 (2007)
38. Yamasaki, K., Gozolchiani, A., Havlin, S.: Climate networks around the globe are significantly affected by el niño. Phys. Rev. Lett. **100**(22), 228501 (2008)

Sensor Scheduling for Space Object Tracking and Collision Alert

Huimin Chen, Dan Shen, Genshe Chen, and Khanh Pham

Abstract Given the increasingly dense environment in both low-earth orbit (LEO) and geostationary orbit (GEO), a sudden change in the trajectory of any existing resident space object (RSO) may cause potential collision damage to space assets. With a constellation of EO/IR sensor platforms and ground radar surveillance systems, it is important to design optimal estimation algorithm for updating nonlinear object states and allocating sensing resources to effectively avoid collisions among many RSOs. We consider N space objects being observed by M sensors whose task is to provide the minimum mean square estimation error of the overall system subject to the cost associated with each measurement. To simplify the analysis, we assume that sensors can switch between objects instantaneously subject to additional resource and sensing geometry constraints. We first formulate the sensor scheduling problem using the optimal control formalism and then derive a tractable relaxation of the original optimization problem, which provides a lower bound on the achievable performance. We propose an open-loop periodic switching policy whose performance can approach the theoretical lower bound closely. We also discuss a special case of identical sensors and derive an index policy that coincides with the general solution to restless multi-armed bandit problem by Whittle. Finally, we demonstrate the effectiveness of the resulting sensor management scheme for space situational awareness using a realistic space object tracking simulator with

H. Chen
Department of Electrical Engineering, University of New Orleans, 2000 Lakeshore Drive, New Orleans, LA 70148, USA
e-mail: hchen2@uno.edu

D. Shen • G. Chen
I-Fusion Technology, Inc., 14163 Furlong Way, Germantown, MD 20874, USA
e-mail: dshen@i-fusion-i.com; gchen@i-fusion-i.com

K. Pham (✉)
AFRL/RVSV, Bernalillo, NM 87117, USA
e-mail: AFRL.RVSV@kirtland.af.mil

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_9,

both unintentional and intentional maneuvers by RSOs that may lead to collision. Our sensor scheduling scheme outperforms the conventional information gain and covariance control based schemes in the overall tracking accuracy as well as making earlier declaration of collision events.

Keywords Sensor management • Sensor scheduling • Nonlinear filtering • Kalman filter • Restless multi-armed bandit • Space object tracking • Collision alert • Situational awareness

1 Introduction

Over recent decades, the space environment has become more complex with a significant increase in space debris among densely populated satellites. Efficient and reliable space operations rely heavily on the space situational awareness where searching and tracking space objects and identifying their intent are crucial in creating a consistent global picture. The development of sensor resource allocation algorithms is crucial for precision tracking and collision alert where a large number of sensors have to monitor a large number of objects. Without loss of generality, we consider M sensors tracking the state of N objects in continuous time. The objects have independent dynamic motion model given by

$$\dot{x}_i = f(x_i, u_i) + w_i, \ i = 1, \ldots, N. \tag{1}$$

We assume that the control inputs $u_i(t)$ are deterministic and known for $t \geq 0$. The process noises $w_i(t)$ are zero mean white Gaussian with known power spectrum density W_i, i.e., $\mathrm{Cov}(w_i(t_0), w_i(t_1)) = W_i \delta(t_0 - t_1), \forall t_0, t_1$. The initial states $x_i(0)$ are independent random variables with known distributions. Both initial states and process noises are mutually independent. If sensor j is used to observe object i, we have the measurement equation given by

$$y_{ij} = h_{ij}(x_i) + v_{ij}, \ i = 1, \ldots, N, j = 1, \ldots, M. \tag{2}$$

The measurement noise v_{ij} is a white Gaussian process with known power spectrum density V_{ij}. Define the indicator variable

$$\pi_{ij}(t) = \begin{cases} 1, \text{ if object } i \text{ is observed by sensor } j \text{ at time } t \\ 0, \text{ otherwise.} \end{cases}$$

We assume that each sensor can observe at most one object at any time instant, i.e.,

$$\sum_{i=1}^{N} \pi_{ij}(t) \leq 1, \ \forall t, \ j = 1, \ldots, M. \tag{3}$$

Denote the sensor scheduling policy $\pi(t) = \{\pi_{ij}(t)\}$. Let $\hat{x}_{\pi,i}$ be the state estimate of x_i under the policy π. The optimal policy is the one that minimizes the average cost over infinite time horizon given by

$$\pi^* = \arg \min_{\pi, \{\hat{x}_{\pi,i}\}} \lim_{T\to\infty} \frac{1}{T} E\left[\int_0^T \sum_{i=1}^{N} \left((\hat{x}_{\pi,i} - x_i)' C_i (\hat{x}_{\pi,i} - x_i) + \sum_{j=1}^{M} \kappa_{ij} \pi_{ij} \right) \mathrm{d}t \right], \tag{4}$$

where C_i's are positive definite weighting matrices and κ_{ij}'s are sensing cost per unit time when sensor j is scheduled to observe object i. Note that the policy π itself can depend on the past observations in a causal manner.

There exist various formulations of sensor scheduling for multi-site surveillance [5], agile sensing [24], and sensor network applications [31]. Conventional sensor-to-object assignment problem is often formulated as one-step look ahead or finite horizon scheduling where the sensing cost serves as a constraint rather than part of the objective function. One popularly used criterion for sensor selection is the total information gain from all the objects being tracked [23]. However, this criterion does not prioritize the objects with respect to their types or identities. Nor does it take time value into account. Alternatively, covariance control optimizes the sensing resources to achieve the desired estimation error covariance for each object [22]. It has the flexibility to design the desired tracking accuracy according to the importance of each object. This may include the assessment of collision probability among those junctions between an object being tracked and a known resident space object (RSO). However, it is unclear a priori whether the desired error covariance is achievable at the scheduled time. Nevertheless, the human operator has to determine the importance of each object in terms of its maximal allowable error covariance. Our sensor scheduling formulation can be converted into maximizing the information gain by letting $C_i = I$ and $\kappa_{ij} = 0$. The policy π can also be associated with non-optimal state estimator provided the estimation error covariance can be faithfully assessed in (4). In addition, the objective function explicitly considers the balance between sensing cost and long-term reward, which has the optimal control flavor. The solution to (4) for the linear case brings valuable insights into nonlinear filtering problem where the optimal sensor schedule is computationally intractable. The rest of the chapter is organized as follows. Section 2 discusses the sensor selection problem in the identical sensor case and connects the solution to the indexability of the restless bandit problem. Section 3 provides the performance lower bound of the general sensor schedule problem and a periodic switching policy that can approach the bound in the linear dynamic case with fast switching frequency of the sensor-to-object assignment. The extension to nonlinear state estimation is also discussed. Simulation of multisensor space object tracking for collision alert is in Sect. 4. Concluding remarks are in Sect. 5.

2 Index-Based Policy

2.1 Optimal State Estimation

For a given policy π, we have the following minimum mean square error estimation problem

$$\min_{\hat{x}_{\pi,i}} \lim_{T\to\infty} \frac{1}{T} E\left[\int_0^T (\hat{x}_{\pi,i} - x_i)' C_i (\hat{x}_{\pi,i} - x_i) \mathrm{d}t\right], i = 1, \ldots, N. \tag{5}$$

The solution to the above nonlinear filtering problem is a conditional mean estimator [30]

$$\begin{aligned} \dot{\hat{x}}_i &= E[f(x_i, u_i)|\{y_{ij}\}, \pi] \\ &\quad + \sum_{j=1}^{M} E[\tilde{x}_i h_{ij}(x_i)'|\{y_{ij}\}, \pi] S_{ij}^{-1} (y_{ij} - E[h_{ij}(x_i)|\{y_{ij}\}, \pi]), \end{aligned} \tag{6}$$

where $\tilde{x}_i = x_i - \hat{x}_i$ and S_{ij} is the innovation matrix given by

$$S_{ij} = E\left[(y_{ij} - E[h_{ij}(x_i)|\{y_{ij}\}, \pi])(y_{ij} - E[h_{ij}(x_i)|\{y_{ij}\}, \pi])'\right]. \tag{7}$$

To gain additional insight on the above solution, we consider linear dynamic model

$$\dot{x}_i = A_i x_i + B_i u_i + w_i \tag{8}$$

and linear measurement equation

$$y_{ij} = H_{ij} x_i + v_{ij}. \tag{9}$$

Then the minimum mean square error filter becomes the celebrated Kalman–Bucy filter given by [1]

$$\dot{\hat{x}}_i = A_i \hat{x}_i + B_i u_i + \Sigma_i \left(\sum_{j=1}^{M} \pi_{ij} H_{ij}' V_{ij}^{-1} (y_{ij} - H_{ij} \hat{x}_i)\right) \tag{10}$$

with the estimation error covariance matrix Σ_i governed by the Riccati equation

$$\dot{\Sigma}_i = A_i \Sigma_i + \Sigma_i A_i' + W_i - \Sigma_i \left(\sum_{j=1}^{M} \pi_{ij} H_{ij}' V_{ij}^{-1} H_{ij}\right) \Sigma_i. \tag{11}$$

Note that the initial condition $\Sigma_i(0)$ depends on the state estimate $\hat{x}_i(0)$ while the dynamics of Σ_i does not depend on the actual observations. Since

$$E[(\hat{x}_{\pi,i} - x_i)' C_i (\hat{x}_{\pi,i} - x_i)] = \mathrm{Tr}(C_i \Sigma_i),$$

we can rewrite the sensor scheduling problem as

$$\pi^* = \arg\min_{\pi} \lim_{T\to\infty} \frac{1}{T} \left[\int_0^T \sum_{i=1}^{N} \left(\mathrm{Tr}(C_i \Sigma_i) + \sum_{j=1}^{M} \kappa_{ij} \pi_{ij} \right) \mathrm{d}t \right], \tag{12}$$

where the estimation of each object achieves the minimum mean square error.

2.2 Multi-armed Bandit Problem

Consider a simplified sensor scheduling problem where all sensors are identical, i.e., $H_{ij} = H_i$, $V_{ij} = V_i$, and $\kappa_{ij} = \kappa_i$, for $j = 1, \ldots, M$. We also introduce a new indicator variable

$$\bar{\pi}_i(t) = \begin{cases} 1, \text{ if object } i \text{ is observed by a sensor at time } t \\ 0, \text{ otherwise.} \end{cases}$$

Without loss of generality, we assume $M \leq N$ and an object can be observed by at most one sensor at any time, i.e.,

$$\sum_{i=1}^{N} \bar{\pi}_i(t) \leq M, \ \forall t. \tag{13}$$

A relaxed version of the sensor scheduling problem can be written as

$$\bar{\pi}^* = \arg\min_{\bar{\pi}} \lim_{T\to\infty} \frac{1}{T} \int_0^T \sum_{i=1}^{N} [\mathrm{Tr}(C_i \Sigma_i) + \kappa_i \bar{\pi}_i] \, \mathrm{d}t \tag{14}$$

subject to

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T \sum_{i=1}^{N} \bar{\pi}_i \, \mathrm{d}t \leq M, \tag{15}$$

where the constraint is only enforced on time average so that we can form the Lagrangian function

$$L(\bar{\pi}, \lambda) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \sum_{i=1}^{N} [\mathrm{Tr}(C_i \Sigma_i) + (\kappa_i + \lambda)\bar{\pi}_i] \, \mathrm{d}t - \lambda M. \tag{16}$$

Note that the above function satisfies

$$L^* = \inf_{\bar{\pi}} \sup_{\lambda} L(\bar{\pi}, \lambda) = \sup_{\lambda} \inf_{\bar{\pi}} L(\bar{\pi}, \lambda) \tag{17}$$

owing to the duality theorem [2], where the dual function

$$\gamma(\lambda) = \inf_{\bar{\pi}} \lim_{T\to\infty} \frac{1}{T} \int_0^T \sum_{i=1}^{N} [\mathrm{Tr}(C_i \Sigma_i) + (\kappa_i + \lambda)\bar{\pi}_i] \, \mathrm{d}t - \lambda M \tag{18}$$

with

$$L^* = \sup_{\lambda} \gamma(\lambda). \tag{19}$$

For any given Lagrangian multiplier λ, the sensor scheduling for object i becomes an optimal control problem

$$\gamma_i(\lambda) = \inf_{\bar{\pi}_i} \lim_{T\to\infty} \frac{1}{T} \int_0^T \mathrm{Tr}(C_i \Sigma_i) + (\kappa_i + \lambda)\bar{\pi}_i \, \mathrm{d}t \tag{20}$$

decoupled from other objects. The coupling is only through the constraint

$$\gamma(\lambda) = \sum_{i=1}^{N} \gamma_i(\lambda) - \lambda M. \tag{21}$$

If we can solve the relaxed sensor scheduling problem, then L^* provides a lower bound of the achievable average cost with hard sensor-to-object assignment constraint.

The above sensor scheduling problem has a close relationship with the classical multi-armed bandit (MAB) problem [16, 33]. In the MAB, we have N arms evolving independently and at most M of them can be activated at any time. Arms that are active will have different dynamics from arms that remain passive. The goal is to find a good policy so that the infinite horizon time averaged reward is maximized. Here arms correspond to objects and their activation corresponds to making an observation by a sensor. Since both active and passive rewards are state dependent, the problem can be viewed as a special case of the restless bandit problem (RBP). Solving the general RBP is PSPACE hard [28]. In fact, attaining any nontrivial approximation factor to the optimal policy of the RBP is also computationally intractable [18]. Fortunately, there are cases of RBP that are indexable, i.e., at any time, each arm can be assigned with an index and the optimal policy activates arms according to their index values [33]. If we examine the objective function (20) carefully, we can see that the passive action becomes more attractive as λ increases. For fixed λ, define $\mathcal{P}_i$ to be the set of values of Σ_i such that $\bar{\pi}_i = 0$ is optimal. The object i is indexable if $\mathcal{P}_i$ expands monotonously

as λ increases. The sensor scheduling problem is said to be indexable if every object is indexable. When object i is indexable, its Whittle index is given by [33]

$$\lambda_i(\Sigma_i) = inf_\lambda\{\Sigma_i \in \mathcal{P}_i(\lambda)\}. \tag{22}$$

Whittle index can be interpreted as the intrinsic value for the measurement of object i, taking into account both the immediate and future gains in reducing the state estimation error. It acts like measurement tax that will make a sensor indifferent between measuring or not measuring the object in terms of the time-averaged cost. At any time, the M objects with the highest current index values will be observed and the performance of this index-based policy is asymptotically optimal as $N \to \infty$ while M/N remains a constant for any indexable RBP [27].

3 Approximate Solution to General Sensor Scheduling Problem

3.1 Lower Bounding the Average Cost

For the scheduling of nonidentical sensors, we first consider the linear dynamics where the Kalman–Bucy filter can be applied to the state estimation of each object. The optimal policy should be the solution to the following optimal control problem

$$C^* = \min_{\pi} \lim_{T\to\infty} \frac{1}{T} \int_0^T \sum_{i=1}^{N} \left[\mathrm{Tr}(C_i \Sigma_i) + \sum_{j=1}^{M} \kappa_{ij} \pi_{ij} \right] \mathrm{d}t \tag{23}$$

subject to

$$\dot{\Sigma}_i = A_i \Sigma_i + \Sigma_i A_i' + W_i - \Sigma_i \left(\sum_{j=1}^{M} \pi_{ij} H_{ij}' V_{ij}^{-1} H_{ij} \right) \Sigma_i, \ i = 1, \ldots, N, \tag{24}$$

$$\sum_{i=1}^{N} \pi_{ij}(t) \le 1, \ \forall t \ge 0, j = 1, \ldots, M, \tag{25}$$

$$\pi_{ij}(t) \in \{0, 1\}, \ \forall t \ge 0, \ i = 1, \ldots, N, \ j = 1, \ldots, M. \tag{26}$$

However, it is unclear whether one can solve the above optimization problem with integer constraints on $\{\pi_{ij}\}$. If we relax the integer constraint, the resulting semidefinite program yields the optimal solution that provides a lower bound of the solution to (23).

Theorem 1. *The average cost obtained by the following semidefinite program*

$$C_l^* = \min_{R_i, Q_i, \bar{\pi}_{ij}} \sum_{i=1}^{N} \left[\mathrm{Tr}(C_i R_i) + \sum_{j=1}^{M} \kappa_{ij} \bar{\pi}_{ij} \right] \tag{27}$$

subject to

$$\begin{bmatrix} R_i & I \\ I & Q_i \end{bmatrix} \geq 0, Q_i > 0, i = 1, \ldots, N, \tag{28}$$

$$\begin{bmatrix} Q_i A_i + A_i' Q_i - \sum_{j=1}^{M} \bar{\pi}_{ij} H_{ij}' V_{ij}^{-1} H_{ij} & Q_i W_i^{1/2} \\ W_i^{1/2} Q_i & -I \end{bmatrix} \leq 0, i = 1, \ldots, N, \tag{29}$$

$$\sum_{i=1}^{N} \bar{\pi}_{ij} \leq 1, \ j = 1, \ldots, M, \tag{30}$$

$$0 \leq \bar{\pi}_{ij} \leq 1, i = 1, \ldots, N, j = 1, \ldots, M \tag{31}$$

provides a lower bound on the achievable cost C^* in (23) for the original problem.

Proof. Let

$$\bar{\pi}_{ij}(t) = \frac{1}{t} \int_0^t \pi_{ij}(\tau) \mathrm{d}\tau. \tag{32}$$

Since $\pi_{ij}(\tau) \in \{0, 1\}$, $\forall \tau \geq 0$, we have $0 \leq \bar{\pi}_{ij}(t) \leq 1$, $\forall t \geq 0$. Define the information matrix $Q_i = \Sigma_i^{-1}$. The dynamics of Q_i satisfies the Riccati equation

$$\dot{Q}_i = -Q_i A_i - A_i' Q_i - Q_i W_i Q_i + \sum_{j=1}^{M} \pi_{ij} H_{ij}' V_{ij}^{-1} H_{ij}, \ i = 1, \ldots, N. \tag{33}$$

Define the time-averaged error covariance and information matrix

$$\bar{\Sigma}_i(t) = \frac{1}{t} \int_0^t \Sigma(\tau) \mathrm{d}\tau, \bar{Q}_i(t) = \frac{1}{t} \int_0^t Q_i(\tau) \mathrm{d}\tau. \tag{34}$$

Owing to the linearity of the trace operator, (23) becomes

$$\min_{\bar{\pi}} \lim_{T \to \infty} \sum_{i=1}^{N} \left[\mathrm{Tr}(C_i \bar{\Sigma}_i(T)) + \sum_{j=1}^{M} \kappa_{ij} \bar{\pi}_{ij}(T) \right]. \tag{35}$$

By invoking Jensen's inequality [4], we have

$$\bar{Q}_i(t) = \frac{1}{t} \int_0^t Q_i(\tau) \tau \geq \left(\frac{1}{t} \int_0^t \Sigma_i(\tau) \mathrm{d}\tau \right)^{-1}, \forall t. \tag{36}$$

Hence

$$\bar{\Sigma}_i(t) \geq \left(\bar{Q}_i(t)\right)^{-1}, \mathrm{Tr}(C_i\,\bar{\Sigma}_i(t)) \geq \mathrm{Tr}(C_i(\bar{Q}_i(t))^{-1}), \forall t. \tag{37}$$

Denote by $Q_i(0) = \Sigma_i(0)^{-1}$. By integrating the Riccati equation, we have

$$\frac{1}{T}[Q_i(T) - Q_i(0)] = -\bar{Q}_i(T)A_i - A_i'\bar{Q}_i(T) - \frac{1}{T}\int_0^T Q_i(t)W_iQ_i(t)\mathrm{d}t + \sum_{j=1}^{M}\left(\frac{1}{T}\int_0^T \pi_{ij}(t)\mathrm{d}t\right)H_{ij}'V_{ij}^{-1}H_{ij},$$

which can be written as

$$\frac{1}{T}[Q_i(T) - Q_i(0)] = -\bar{Q}_i(T)A_i - A_i'\bar{Q}_i(T) - \frac{1}{T}\int_0^T Q_i(t)W_iQ_i(t)\mathrm{d}t + \sum_{j=1}^{M}\bar{\pi}_{ij}(T)H_{ij}'V_{ij}^{-1}H_{ij}.$$

Using Jensen's inequality again, we have

$$\frac{1}{T}\int_0^T Q_i(t)W_iQ_i(t)\mathrm{d}t \geq \bar{Q}_i(T)W_i\bar{Q}_i(T), \tag{38}$$

so

$$\bar{Q}_i(T)A_i + A_i'\bar{Q}_i(T) + \bar{Q}_i(T)W_i\bar{Q}_i(T) - \sum_{j=1}^{M}\bar{\pi}_{ij}(T)H_{ij}'V_{ij}^{-1}H_{ij} \leq \frac{Q_i(0)}{T}, \forall T. \tag{39}$$

We can see that for any given policy π and at any time T, the quantity

$$\sum_{i=1}^{N}\left[\mathrm{Tr}(C_i\,\bar{\Sigma}_i(T)) + \sum_{j=1}^{M}\kappa_{ij}\bar{\pi}_{ij}(T)\right]$$

is bounded below by

$$\sum_{i=1}^{N}\left[\mathrm{Tr}(C_i(\bar{Q}_i(T))^{-1}) + \sum_{j=1}^{M}\kappa_{ij}\bar{\pi}_{ij}(T)\right]$$

with the constraint (39) for matrices $\bar{Q}_i(T)$ and

$$0 \leq \bar{\pi}_{ij}(T) \leq 1, i = 1, \ldots, N, j = 1, \ldots, M, \tag{40}$$

$$\sum_{i=1}^{N} \bar{\pi}_{ij}(T) \leq 1, j = 1, \ldots, M \tag{41}$$

for $\bar{\pi}_{ij}(T)$. Taking $T \to \infty$, we have the lower bound of the achievable cost given by

$$\min_{Q_i, \bar{\pi}_{ij}} \sum_{i=1}^{N} \left[\mathrm{Tr}(C_i Q_i^{-1}) + \sum_{j=1}^{M} \kappa_{ij} \bar{\pi}_{ij} \right] \tag{42}$$

subject to

$$Q_i A_i + A_i' Q_i + Q_i W_i Q_i - \sum_{j=1}^{M} \bar{\pi}_{ij} H_{ij}' V_{ij}^{-1} H_{ij} \leq 0, Q_i > 0, i = 1, \ldots, N, \tag{43}$$

$$\sum_{i=1}^{N} \bar{\pi}_{ij} \leq 1, \geq 0, j = 1, \ldots, M, \tag{44}$$

$$0 \leq \bar{\pi}_{ij} \leq 1, i = 1, \ldots, N, j = 1, \ldots, M. \tag{45}$$

By introducing the slack variable R_i such that $R_i \geq Q_i^{-1}$, the problem (42) can be written as the semidefinite program given by (27) with constraints (28)–(31). □

3.2 *Approximate Solution that Approaches the Lower Bound*

By solving the semidefinite program (27), we can obtain the lower bound of the average cost (23) in the original sensor scheduling problem. The resulting $\{\bar{\pi}_{ij}\}$ may take non-integer values within $[0, 1]$ and Q_i can be interpreted as the optimal long-term information matrix of object i corresponding to the maximum Fisher information that one would hope to obtain by utilizing the available sensing resources. Denote by $\bar{\Pi} = [\bar{\pi}_{ij}]$ the sensor-to-object association matrix that solves the semidefinite program (27). We want to decompose $\bar{\Pi}$ to a linear combination of feasible sensor-to-object assignment matrices.

Theorem 2. *For any matrix $\bar{\Pi}$ that satisfies*

$$\sum_{i=1}^{N} \bar{\pi}_{ij} \leq 1, j = 1, \ldots, M, 0 \leq \bar{\pi}_{ij} \leq 1, i = 1, \ldots, N, j = 1, \ldots, M, \tag{46}$$

there exists some integer K such that

$$\bar{\Pi} = \sum_{k=1}^{K} w_k P_k \tag{47}$$

with

$$w_k > 0, k = 1, \ldots, K \tag{48}$$

and $\{P_k\}$'s are feasible sensor-to-object assignment matrices.

Proof. For $M = 1$, if $\bar{\pi}_{i1} \neq 0$, we can set $w_i = \bar{\pi}_{i1}$ and P_i has a single nonzero entry indicating that object i will be observed. When $M > 1$, we need to expand the basis of fundamental object-to-sensor assignment matrices. A general way to handle this situation is by introducing dummy sensor and dummy object. With dummy sensors that have infinite noise power spectrum density, we can make $\bar{\Pi}$ an $N \times N$ doubly sub-stochastic matrix where $N - M$ additional columns of zeros are added. Using Birkhoff theorem, we may decompose $\bar{\Pi}$ as a convex combination of permutation matrices [3]. Thus we have the feasible sensor-to-object assignment decomposition. In fact, there exists efficient algorithm that finds the weights $\{w_k\}$ in $O(N^{4.5})$ for $K = O(N^2)$ coefficients [8]. □

Now we can design a sensor scheduling policy that periodically switches among the sensor-to-object assignment matrices $\{P_k\}$ to approximate the optimal solution to (23). Let δ be some duration of time. At time $t = 0$, the sensor-to-object assignment is based on P_1, i.e., we allow sensor j to observe object i only when $p_{1,i,j} = 1$. Once the observations are made, each object will update its state estimate according to the Kalman–Bucy filter. At time $t = w_1\delta$, we switch the sensor schedule according to P_2. At time $t = (w_1 + w_2)\delta$, we switch the sensor schedule according to P_3 and so on. Note that the sensor schedule will go back to P_1 after a period of δ.

Theorem 3. *Let C_δ^* be the average cost associated with the periodic switching policy π_δ among the sensor-to-object assignment matrices $\{P_k\}$. Then $C_\delta^* - C_l^* = o(\delta)$ as $\delta \to 0$.*

Proof. Denote by $\Sigma_i^\delta(t)$ the estimation error covariance of object i under the periodic sensor scheduling policy π_δ. When the whole state estimator reaches to its steady state as $t \to \infty$, $\Sigma_i^\delta(t)$ will converge to a periodic function $\bar{\Sigma}_i^\delta(t)$ with period δ. The matrix $\bar{\Sigma}_i^\delta(t)$ satisfies the periodic Riccati equation

$$\dot{\bar{\Sigma}}_i^\delta = A_i \bar{\Sigma}_i^\delta + A_i' \bar{\Sigma}_i^\delta + W_i - \bar{\Sigma}_i^\delta (H_i^\delta)' H_i^\delta \bar{\Sigma}_i^\delta \tag{49}$$

with initial condition $\bar{\Sigma}_i^\delta(0) = \Sigma_i(0)$ where

$$H_i^\delta = \sum_{j=1}^{M} V_{ij,\pi_\delta}^{-1/2} H_{ij,\pi_\delta} \tag{50}$$

is a piecewise constant value function with period δ [7]. Let $\bar{\Sigma}_i^{\delta}$ be the average of $\bar{\Sigma}_i^{\delta}(t)$

$$\bar{\Sigma}_i^{\delta} = \frac{1}{\delta}\int_0^{\delta} \bar{\Sigma}_i^{\delta}(\tau)\mathrm{d}\tau \tag{51}$$

and we can see that $\bar{\Sigma}_i^{\delta}(t) - \bar{\Sigma}_i^{\delta} = o(\delta)$, $\forall t$. As $\delta \to 0$, $\bar{\Sigma}_i^{\delta}$ converges to the unique solution to the algebraic Riccati equation [1]

$$A_i\bar{\Sigma}_i + \bar{\Sigma}_i A_i' + W_i - \bar{\Sigma}_i\left(\sum_{j=1}^{M}\bar{\pi}_{ij}H_{ij}'V_{ij}^{-1}H_{ij}\right)\bar{\Sigma}_i = 0, \tag{52}$$

since

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T\sum_{i=1}^{N}\sum_{j=1}^{M}\kappa_{ij}\pi_{ij}^{\delta}(t)\mathrm{d}t = \sum_{i=1}^{N}\sum_{j=1}^{M}\kappa_{ij}\bar{\pi}_{ij}. \tag{53}$$

When $\bar{\pi}_{ij}$'s are obtained by solving the semidefinite program (27), $\bar{Q}_i = \bar{\Sigma}_i^{-1}$ also minimizes $\sum_{i=1}^{N}\mathrm{Tr}(C_i Q_i^{-1})$ for all matrices $Q_i > 0$ that satisfy

$$Q_i A_i + A_i' Q_i + Q_i W_i Q_i - \sum_{j=1}^{M}\bar{\pi}_{ij}H_{ij}'V_{ij}^{-1}H_{ij} \le 0. \tag{54}$$

Thus the policy π_δ has the resulting $\{\bar{\pi}_{ij}, \bar{Q}_i\}$ with the average cost no greater than $C_l^* + o(\delta)$ as $\delta \to 0$. □

In summary, the optimal average cost $C^* \in [C_l^*, C_l^* + o(\delta)]$. Unfortunately, the M sensors have to switch among K different sensor-to-object assignment solutions with appropriate coverage intervals within a short period δ in order to approach the lower bound C_l^*. The analysis is based on linear dynamic state and measurement equation for each object. We will have to extend the results to the nonlinear estimation case.

3.3 Nonlinear Filter Design and Performance Bound

3.3.1 Recursive Linear Minimum Mean Square Error Filter

When a space object has been detected, a tracking filter will predict the object's state at any time in the future based on the available sensor measurements. Both the state dynamics and measurement equation are nonlinear resulting in the nonlinear state estimator for each object. Despite the abundant literature on nonlinear filter design [6, 11, 12, 14, 19, 25], we chose the following tracking filter

based on our earlier study [9]. With any given sensor schedule policy, we use the following notations for state estimation of any nonlinear dynamic system. Let $\mathbf{x} = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}]'$ be the position and velocity of a space object in the earth-center earth-fixed coordinate system. Denote by $\hat{\mathbf{x}}_k^-$ the state prediction from time t_{k-1} to time t_k based on the state estimate $\hat{\mathbf{x}}_{k-1}^+$ at time t_{k-1} with all measurements up to t_{k-1}. The prediction is made by numerically integrating the state equation given by

$$\dot{\hat{\mathbf{x}}}(t) = f(\hat{\mathbf{x}}(t), u(t)) \tag{55}$$

without the process noise. The mean square error (MSE) of the state prediction is obtained by numerically integrating the following matrix equation:

$$\dot{P}(t) = F(\hat{\mathbf{x}}_k^-)P(t) + P(t)F(\hat{\mathbf{x}}_k^-)^T + W(t), \tag{56}$$

where $F(\hat{\mathbf{x}}_k^-)$ is the Jacobian matrix of the Keplerian orbital dynamics given by

$$F(\mathbf{x}) = \begin{bmatrix} 0_{3\times3} & I_3 \\ F_0(\mathbf{x}) & 0_{3\times3} \end{bmatrix}, \tag{57}$$

$$F_0(\mathbf{x}) = \mu \begin{bmatrix} \frac{3x^2}{r^5} - \frac{1}{r^3} & \frac{3xy}{r^5} & \frac{3xz}{r^5} \\ \frac{3xy}{r^5} & \frac{3y^2}{r^5} - \frac{1}{r^3} & \frac{3yz}{r^5} \\ \frac{3xz}{r^5} & \frac{3yz}{r^5} & \frac{3z^2}{r^5} - \frac{1}{r^3} \end{bmatrix}, \tag{58}$$

$$r = \sqrt{x^2 + y^2 + z^2} \tag{59}$$

and evaluated at $\mathbf{x} = \hat{\mathbf{x}}_k^-$. The sensor measurement $\mathbf{z}_k$ obtained at time t_k is given by

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \tag{60}$$

where

$$\mathbf{v}_k \sim \mathcal{N}(0, V_k) \tag{61}$$

is the measurement noise, which is assumed independent of each other and independent to the initial state as well as process noise.

Let $\mathbf{Z}^k$ be the cumulative sensor measurements up to t_k from a fixed sensor scheduling policy. The recursive linear minimum mean square error (LMMSE) filter applies the following update equation [1]:

$$\hat{\mathbf{x}}_{k|k} \triangleq E^*\left[\mathbf{x}_k|\mathbf{Z}^k\right] = \hat{\mathbf{x}}_{k|k-1} + K_k\tilde{\mathbf{z}}_{k|k-1}, \tag{62}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k', \tag{63}$$

where

$$\hat{\mathbf{x}}_{k|k-1} = E^*\left[\mathbf{x}_k|\mathbf{Z}^{k-1}\right],$$

$$\hat{\mathbf{z}}_{k|k-1} = E^*\left[\mathbf{z}_k|\mathbf{Z}^{k-1}\right],$$

$$\begin{aligned}
\tilde{\mathbf{x}}_{k|k-1} &= \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}, \\
\tilde{\mathbf{z}}_{k|k-1} &= \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}, \\
P_{k|k-1} &= E\left[\tilde{\mathbf{x}}_{k|k-1}\tilde{\mathbf{x}}'_{k|k-1}\right], \\
S_k &= E\left[\tilde{\mathbf{z}}_{k|k-1}\tilde{\mathbf{z}}'_{k|k-1}\right], \\
K_k &= C_{\tilde{\mathbf{x}}_k\tilde{\mathbf{z}}_k} S_k^{-1}, \\
C_{\tilde{\mathbf{x}}_k\tilde{\mathbf{z}}_k} &= E\left[\tilde{\mathbf{x}}_{k|k-1}\tilde{\mathbf{z}}'_{k|k-1}\right].
\end{aligned}$$

Note that $E^*[\cdot]$ becomes the conditional mean of the state for linear Gaussian dynamics and the above filtering equations become the celebrated Kalman filter [1]. For nonlinear dynamic system, (62) is optimal in the mean square error sense when the state estimate is constrained to be an affine function of the measurement. Given the state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ and its error covariance $P_{k-1|k-1}$ at time t_{k-1}, if the state prediction $\hat{\mathbf{x}}_{k|k-1}$, the corresponding error covariance $P_{k|k-1}$, the measurement prediction $\hat{\mathbf{z}}_{k|k-1}$, the corresponding error covariance S_k, and the crosscovariance $E\left[\tilde{\mathbf{x}}_{k|k-1}\tilde{\mathbf{z}}'_{k|k-1}\right]$ in (62) and (63) can be expressed as a function only through $\hat{\mathbf{x}}_{k-1|k-1}$ and $P_{k-1|k-1}$, then the above formula is truly recursive. However, for general nonlinear system dynamics (1) and measurement equation (60) , we have

$$\hat{\mathbf{x}}_{k|k-1} = E^*\left[\int_{t_{k-1}}^{t_k} f(\mathbf{x}(t), \mathbf{w}(t))\mathrm{d}t + \mathbf{x}_{k-1}|\mathbf{Z}^{k-1}\right], \tag{64}$$

$$\hat{\mathbf{z}}_{k|k-1} = E^*\left[h(\mathbf{x}_k, \mathbf{v}_k)|\mathbf{Z}^{k-1}\right]. \tag{65}$$

Both $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{z}}_{k|k-1}$ will depend on the measurement history $\mathbf{Z}^{k-1}$ and the corresponding moments in the LMMSE formula. In order to have a truly recursive filter, the required terms at time t_k can be obtained *approximately* through $\hat{\mathbf{x}}_{k-1|k-1}$ and $P_{k-1|k-1}$, i.e.,

$$\begin{aligned}
\left\{\hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}\right\} &\approx \mathrm{Pred}\left[f(\cdot), \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}\right], \\
\left\{\hat{\mathbf{z}}_{k|k-1}, S_k, C_{\tilde{\mathbf{x}}_k\tilde{\mathbf{z}}_k}\right\} &\approx \mathrm{Pred}\left[h(\cdot), \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}\right],
\end{aligned}$$

where $\mathrm{Pred}\left[f(\cdot), \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}\right]$ denotes that $\left\{\hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}\right\}$ propagates through the nonlinear function $f(\cdot)$ to approximate $E^*\left[f(\cdot)|\mathbf{Z}^{k-1}\right]$ and the corresponding error covariance $P_{k|k-1}$.

Similarly, $\mathrm{Pred}\left[h(\cdot), \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}\right]$ predicts the measurement and the corresponding error covariance only through the approximated state prediction. This poses difficulties for the implementation of the recursive LMMSE filter due to insufficient information. The prediction of a random variable going through a nonlinear function, most often, cannot be completely determined using only the

first and second moments. Two remedies are often used: One is to approximate the system via unscented transform such that the prediction based on the approximated system can be carried out only through $\{\hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}\}$ [20, 21]. Another is by approximating the density function with a set of particles and propagating those particles in the recursive Bayesian filtering framework, i.e., using a particle filter [13, 15, 17].

3.3.2 Posterior Cramer–Rao Lower Bound of the State Estimation Error

When computing the dynamics of the state estimation error covariance, the sensor scheduler can use the performance bound without requiring to optimize the sensor-to-object assignment with respect to a particularly designed nonlinear state estimator. Denote by $J(t)$ the Fisher information matrix. Then the posterior Cramer–Rao lower bound (PCRLB) is given by [32]

$$B(t) = J(t)^{-1}, \tag{66}$$

which quantifies the ideal mean square error of any filtering algorithm, i.e.,

$$E\left[(\hat{\mathbf{x}}(t_k) - \mathbf{x}(t_k))(\hat{\mathbf{x}}(t_k) - \mathbf{x}(t_k))^T | \mathbf{Z}^k\right] \geq B(t_k). \tag{67}$$

Assuming an additive white Gaussian process noise model, the Fisher information matrix satisfies the following differential equation:

$$\dot{J}(t) = -J(t)F(\mathbf{x}) - F(\mathbf{x})^T J(t) - J(t)Q(t)J(t) \tag{68}$$

for $t_{k-1} \leq t \leq t_k$ where F is the Jacobian matrix given by

$$F(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}. \tag{69}$$

When a measurement is obtained at time t_k with additive Gaussian noise $\mathcal{N}(0, R_k)$, the new Fisher information matrix is

$$J(t_k^+) = J(t_k^-) + E_{\mathbf{x}}\left[H(\mathbf{x})^T R_k^{-1} H(\mathbf{x})\right], \tag{70}$$

where H is the Jacobian matrix given by

$$H(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}. \tag{71}$$

The initial condition for the recursion is $J(t_0)$ and the PCRLB can be obtained with respect to the true distribution of the state $\mathbf{x}(t)$.

In practice, the recursive LMMSE filter will be used to track each space object. The sensor manager will use the estimated state to compute the PCRLB and solve the semidefinite program (27) by replacing Q_i with the Fisher information matrix of object i. The resulting periodic switching policy will have to be updated at the highest sensor revisit rate in order to approach the performance lower bound of the average cost. Alternatively, the sensor manager can apply information gain-based policy or index-based policy which require less computation within a fixed horizon. See [10] for specific sensor management implementations that utilize PCRLB for orbital object tracking.

4 Simulation Study

4.1 Scenario Description

We consider a small-scale space object tracking and collision alert scenario where 30 LEO observers collaboratively track 3 LEO satellites (called red team) and monitor 5 LEO asset satellites (called blue team). The orbital trajectories are created with the same altitude similar to those real satellites from the NORAD catalog, but we can change the orbital trajectories to generate a collision event between an object from the red team and an object from the blue team. The associated tracking errors for each object in the red team were obtained based on the recursive linear minimum mean square error filter when sensors are assigned to objects according to some criterion based on the non-maneuvering motion. We assume that the orbital trajectories of LEO observers and blue team are known to red team. We also assume that each observer can update the sensing schedule no sooner than 50 s. The sensor schedule is based on the weights being proportional to the estimated collision probability over the impact time. The estimation of collision probability and impact time was presented in [26].

Red team may direct an unannounced object to perform intelligent maneuver that changes the inclination of its orbit. In particular, at time $t = 1{,}000$ s, object 1 performs a 1 s burn that produces a specific thrust which leads to a collision to object 3 in the blue team in 785 s. At time $t = 1{,}523$ s, object 2 performs a 1 s burn that produces a specific thrust which leads to a collision event to object 5 in the blue team in 524 s. Note that the maneuver onset time of object 2 is chosen to have the Earth blockage of the closest 3 LEO observers for more than 200 s. The maneuver is also lethal because of the collision path to the closest asset satellite in less than 9 min. Within 1,000 s and 2,000 s, object 3 performs a 1 s burn with random maneuver onset time that does not lead to a collision. The goal of sensor selection is to improve the tracking accuracy and declare the collision event as early as possible with false alarm below a desirable rate. Each observer has range, bearing, elevation, and range rate measurements with standard deviations 100 m, 10 mrad, 10 mrad, and 2 m/s, respectively. We applied the generalized Page's test (GPT) for maneuver

Table 1 Comparison of tracking accuracy and maneuver detection delay

Object	1	2	3
(i) Average delay (s)	126	438	83
(i) Average peak position error (km)	23.4	53.3	13.6
(i) Average peak velocity error (km/s)	0.29	0.38	0.21
(ii) Average delay (s)	133	149	92
(ii) Average peak position error (km)	24.3	26.7	14.8
(ii) Average peak velocity error (km/s)	0.30	0.33	0.23
(iii) Average delay (s)	154	177	101
(iii) Average peak position error (km)	26.1	28.4	16.3
(iii) Average peak velocity error (km/s)	0.32	0.35	0.26

onset detection while the filter update of the state estimate does not use the range rate measurement [29]. The reason is that the nonlinear filter designed assuming non-maneuver target motion is sensitive to the model mismatch in the range rate when a space object maneuvers. The thresholds of the GPT were chosen to have the false alarm probability $P_{\text{FA}} = 1\%$.

4.2 Performance Comparison

We studied three different sensor management (SM) configurations. (i) Information-based method: Sensors are selected with a uniform sampling interval of 50 s to maximize the total information gain. (ii) Periodic switching method: Sensing actions are scheduled to minimize the average cost by solving the semidefinite program (27). (iii) Greedy method: Sensing actions are obtained using Whittle's index obtained assuming identical sensors. We ran 200 Monte Carlo simulations on the tracking and collision alert scenario for each SM configuration and compare both tracking and collision alert performance as opposed to the criteria used in the SM schemes. Table 1 shows the peak errors in position and velocity for each object in red team based on the centralized tracker using the measurements from three different SM schemes. The average detection delays for each object are also shown in Table 1. We can see that both the maneuver detection delay and average peak estimation error are larger using the conventional SM scheme (i) than the ones that optimize the cost over infinite horizon—(ii) and (iii)—for object 2. Note that object 2 has a lethal maneuver that requires more prompt sensing action to make early declaration of the collision event. However, the immediate information gain may not be as large as that from object 1. The covariance control based method will not make any correction to the sensing schedule either before the maneuver detection of object 2, which is a consequence of planning over a short time horizon. Interestingly, the performance degradation is quite mild for the index-based SM scheme compared with its near-optimal counterpart.

Next, we compare the collision detection performance as well as the average time between the collision alert and its occurrence. We also compute the average number

Table 2 Performance comparison of collision detection probability and average early-warning duration

Configuration	Detection probability	False alarm probability	Average duration (s)	Average scans
(i) Object 1	0.88	0.04	514	2.8
(i) Object 2	0.31	0.05	138	2.5
(ii) Object 1	0.93	0.04	518	2.6
(ii) Object 2	0.85	0.03	346	2.2
(iii) Object 1	0.88	0.04	502	2.5
(iii) Object 2	0.78	0.04	328	2.4

of scans required to declare a collision event starting from the maneuver onset time. A collision alert will be declared when the closest encounter of two space objects is within 10 km with at least 99% probability based on the predicted orbital states. The false alarm probability is estimated from the collision declaration occurrence between object 3 and any of the asset satellites. The performance of collision alert with three SM schemes is shown in Table 2. We can see that the information gain-based method (configuration (i)) yields much smaller collision detection probability for object 2. Among those collision declarations for object 2, the average duration between the collision alert and the actual encounter time is much shorter using configuration (i) than using configurations (ii) and (iii). Thus blue team will have limited response time in choosing the appropriate collision avoidance action. This is mainly due to the long delay of detecting maneuvering object thus leading to large tracking error as seen in Table 1. In contrast, periodic switching method (configuration (ii)) achieves much more accurate collision detection with longer early warning time on average. It is worth noting that the index-based method (configuration (iii)) yields slightly worse performance than that of configuration (ii) due to its greedy manner in solving the non-indexable RBP. Nevertheless, configuration (iii) is computationally more efficient and yields satisfactory performance compared with the near-optimal policy (ii).

5 Summary and Conclusions

We studied the sensor scheduling problem where N space objects are monitored by M sensors whose task is to provide the minimum mean square estimation error of the overall system subject to the cost associated with each measurement. We first formulated the sensor scheduling problem using the optimal control formalism and then derive a tractable relaxation of the original optimization problem, which provides a lower bound on the achievable performance. We proposed an open-loop periodic switching policy whose performance is arbitrarily close to the theoretical lower bound. We also discussed a special case of identical sensors and derive an index policy that coincides with the general solution to restless multi-armed bandit

problem by Whittle. Finally, we demonstrated the effectiveness of the resulting sensor management scheme for space situational awareness using a realistic space object tracking scenario with both unintentional and intentional maneuvers by RSOs that may lead to collision. Our sensor scheduling scheme outperforms the conventional information gain and covariance control based schemes in the overall tracking accuracy as well as making earlier declaration of collision events. The index policy has a slight performance degradation than the near-optimal periodic switching policy with reduced computational cost, which seems to be applicable to large-scale problems.

Acknowledgment H. Chen was supported in part by ARO through grant W911NF- 08-1-0409, ONR-DEPSCoR through grant N00014-09-1-1169 and Office of Research & Sponsored Programs at University of New Orleans. The authors are grateful to the anonymous reviewers for their constructive comments to an earlier draft of this work.

References

1. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software. Wiley, New York (2001)
2. Bertsekas, D.: Dynamic Programming and Optimal Control (2nd edn.). Athena Scientific, Belmont (2001)
3. Birkhoff, G.: Tres observaciones sobre el algebra lineal. Univ. Nac. Tucuman Rev. **5**, 147–151 (1946)
4. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
5. Boyko, N., Turko, T., Boginski, V., Jeffcoat, D.E., Uryasev, S., Zrazhevsky, G., Pardalos, P.M.: Robust multi-sensor scheduling for multi-site surveillance. J. Comb. Optim. **22**(1), 35–51 (2011)
6. Carme, S., Pham, D.-T., Verron, J.: Improving the singular evolutive extended Kalman filter for strongly nonlinear models for use in ocean data assimilation. Inverse Probl. **17**, 1535–1559 (2001)
7. Carpanese, N.: Periodic Riccati difference equation: approaching equilibria by implicit systems. IEEE Trans. Autom. Contr. **45(7)**, 1391–1396 (2000)
8. Chang, C., Chen, W., Huang, H.: Birkhoff-von Neumann input buffered crossbar switches. In: Proc. IEEE INFORCOM. **3**, 1614–1623 (2000)
9. Chen, H., Chen, G., Blasch, E.P., Pham, K.: Comparison of several space target tracking filters. In: Proceedings of SPIE Defense, Security Sensing, vol. 7730, Orlando (2009)
10. Chen, H., Chen, G., Shen, D., Blasch, E.P., Pham, K.: Orbital evasive target tracking and sensor management. In: Dynamics of Information Systems: Theory and Applications. Hirsch, M.J., Pardalos, P.M., Murphey, R. (eds.), Lecture Notes in Control and Information Sciences. Springer, New York (2010)
11. Daum, F.E.: Exact finite-dimensional nonlinear filters. IEEE Trans. Autom. Contr. **31**, 616–622 (1986)
12. Daum, F.E.: Nonlinear filters: beyond the Kalman filter. IEEE Aerosp. Electron. Syst. Mag. **20**, 57–69 (2005)
13. Doucet, A., de Frietas, N., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science. Springer, New York (2001)
14. Evensen, G.: Data Assimilation: The Ensemble Kalman Filter. Springer, New York (2006)

15. Gilks, W.R., Berzuini, C.: Following a moving target—Monte Carlo inference for dynamic Bayesian models. J. R. Stat. Soc. B **63**, 127–146 (2001)
16. Gittins, J.C., Jones, D.M.: A dynamic allocation index for the sequential design of experiments. In: Progress in Statistics (European Meeting of Statisticians) (1972)
17. Gordon, N., Salmond, D., Smith, A.F.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc. F **140(2)**, 107–113 (1993)
18. Guha, S., Munagala, K.: Approximation algorithms for budgeted learning problems. In: Proceedings ACM Symposium on Theory of Computing (2007)
19. Houtekamer, P.L., Mitchell, H.L.: Data assimilation using an ensemble Kalman filter technique. Monthly Weather Rev. **126**, 796–811 (1998)
20. Julier, S., Uhlmann, J., Durrant-Whyte, H.F.: A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Trans. Autom. Contr. **45**, 477–482 (2000)
21. Julier, S., Uhlmann, J.: Unscented filtering and nonlinear estimation. Proc. IEEE **92(3)**, 401–422 (2004)
22. Kalandros, M., Pao, L.Y.: Covariance control for multisensor systems. IEEE Trans. Aerosp. Electron. Syst. **38**, 1138–1157 (2002)
23. Kreucher, C.M., Hero, A.O., Kastella, K.D., Morelande, M.R.: An information based approach to sensor management in large dynamic networks. Proc. IEEE **95**, 978–999 (2007)
24. Lemaitre, M., Verfaille, G., Jouhaud, F., Lachiver, J.M., Bataille N.: Selecting and scheduling observations of agile satellites. Aerosp. Sci. Technol. **6**, 367–381 (2002)
25. Li, X.R., Jilkov, V.P.: A survey of maneuvering target tracking: approximation techniques for nonlinear filtering. In: Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, vol. 5428–62, Orlando (2004)
26. Maus, A., Chen, H., Oduwole, A., Charalampidis, D.: Designing collision alert system for space situational awareness. In: 20th ANNIE Conference, St. Louis, MO (2010)
27. Nino-Mora, J.: Restless bandits, partial conservation laws and indexability. Adv. Appl. Prob. **33**, 76–98 (2001)
28. Papadimitriou, C., Tsitsiklis, J.: The complexity of optimal queueing network control. Math. Oper. Res. **2**, 293–305 (1999)
29. Ru, J., Chen, H., Li, X.R., Chen, G.: A range rate based detection technique for tracking a maneuvering target. In: Proceedings of SPIE Conference on Signal and Data Processing of Small Targets (2005)
30. Sage, A., Melsa, J.: Estimation Theory with Applications to Communications and Control. McGraw-Hill, USA (1971)
31. Sorokin, A., Boyko, N., Boginski, V., Uryasev, S., Pardalos, P.M.: Mathematical programming tehcniques for sensor networks. Algorithms **2**, 565–581 (2009)
32. Van Trees, H.L.: Detection, Estimation, and Modulation Theory, Part I. Wiley, New York (1968)
33. Whittle, P.: Restless bandits: Activity allocation in a changing world. J. Appl. Probab. **25**, 287–298 (1988)

Throughput Maximization in CSMA Networks with Collisions and Hidden Terminals

Sankrith Subramanian, Eduardo L. Pasiliao, John M. Shea, Jess W. Curtis, and Warren E. Dixon

Abstract The throughput at the medium-access control (MAC) layer in a wireless network that uses the carrier-sense multiple-access (CSMA) protocol is degraded by collisions caused by failures of the carrier-sensing mechanism. Two sources of failure in the carrier-sensing mechanism are delays in the carrier-sensing mechanism and hidden terminals, in which an ongoing transmission cannot be detected at a terminal that wishes to transmit because the path loss from the active transmitter is large. In this chapter, the effect of these carrier-sensing failures is modeled using a continuous-time Markov model. The throughput of the network is determined using the stationary distribution of the Markov model. The throughput is maximized by finding optimal mean transmission rates for the terminals in the network subject to constraints on successfully transmitting packets at a rate that is at least as great as the packet arrival rate.

Keywords Medium access control • Carrier-sense multiple access • CSMA Markov chain • Throughput • Convex optimization

S. Subramanian (✉) • J.M. Shea • W.E. Dixon
Department of Electrical and Computer Engineering, University of Florida, Gainesville FL 32611, USA
e-mail: sankrith@ufl.edu; jshea@ece.ufl.edu; wdixon@ufl.edu

E.L. Pasiliao • J.W. Curtis
Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542, USA
e-mail: pasiliao@eglin.af.mil; curtisjw@eglin.af.mil

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_10,

1 Introduction

Quality of servise (QoS) management and throughput maximization are important capabilities for tactical and mission-critical wireless networks. In the last few years, most research efforts in this area have focused on the optimization and control of specific layers in the communications stack. Examples include specialized QoS-enabled middleware, as well as protocols and algorithms at the transport, network, data link and physical layers. In this work, an analytical framework that allows optimization of the MAC protocol transmission rates in the presence of collisions is developed that will enable further work on cross-layer design involving the MAC layer.

MAC layer throughput optimization focuses on manipulating specific parameters of the MAC layer, including window sizes and transmission rates to maximize/optimize the throughput in the presence of constraints. MAC protocols have been the focus of wireless networks research for the last several years. For example, the use of Markov chains was introduced in [5, 14] to analyze the performance of carrier-sense multiple access (CSMA) MAC algorithms. Performance and throughput analysis of the conventional binary exponential backoff algorithms have been investigated in [3, 4]. In most cases, previous MAC-level optimization algorithms have focused primarily on parameters and feedback from the MAC layer by excluding collisions during the analysis (cf. [5, 10]). In this chapter, we introduce and discuss an approach to include collisions in mobile ad hoc networks for MAC optimization.

Preliminary work on CSMA throughput modeling and analysis was done in [5] based on the assumption that the propagation delay between neighboring nodes is zero. A continuous Markov model was developed to provide the framework and motivation for developing an algorithm that maximizes throughput in the presence of propagation delay. In [10], a collision-free model is used to quantify and optimize the throughput of the network. The feasibility of the arrival rate vector guarantees the reachability of maximum throughput, which in turn satisfies the constraint that the service rate is greater than or equal to the arrival rate, assuming that the propagation delay is zero. In general, the effects of propagation delay play a crucial role on the behavior and throughput of a communication network. Recent efforts attempted various strategies to include delay models in the throughput model. For example, in [13], delay is introduced, and is used to analyze and characterize the achievable rate region for static CSMA schedulers. Collisions, and hence delays, are incorporated in [9] in the Markov model. The mean transmission length of the packets is used as the control variable to maximize the throughput. In this chapter, a model for propagation delay is proposed and incorporated in the model for throughput, and the latter is optimized by first formulating an unconstrained problem and then a constrained problem in the presence of practical rate constraints in the network. Instead of mean transmission lengths (cf. [9]), these formulations are solved using an appropriate numerical optimization technique to obtain the optimal mean transmission rates.

This chapter introduces a throughput model based on [10]. A continuous-time CSMA Markov chain is used to capture the MAC layer dynamics, and the collisions in the network are modeled based on the influence of adjacent links. The waiting times are independently and exponentially distributed. Collisions due to hidden terminals in the network are also modeled and analyzed. Link throughput is optimized by optimizing the waiting times in the network.

2 Network Model

Consider an $(n + k)$-link network with $n + k + 1$ nodes as shown in Fig. 1, where network A consists of n links and network B consists of k links. Assume that all nodes can sense all other nodes in the network. However, there is a sensing delay, so that if two nodes initiate packet transmission within a time duration of δT_s, there

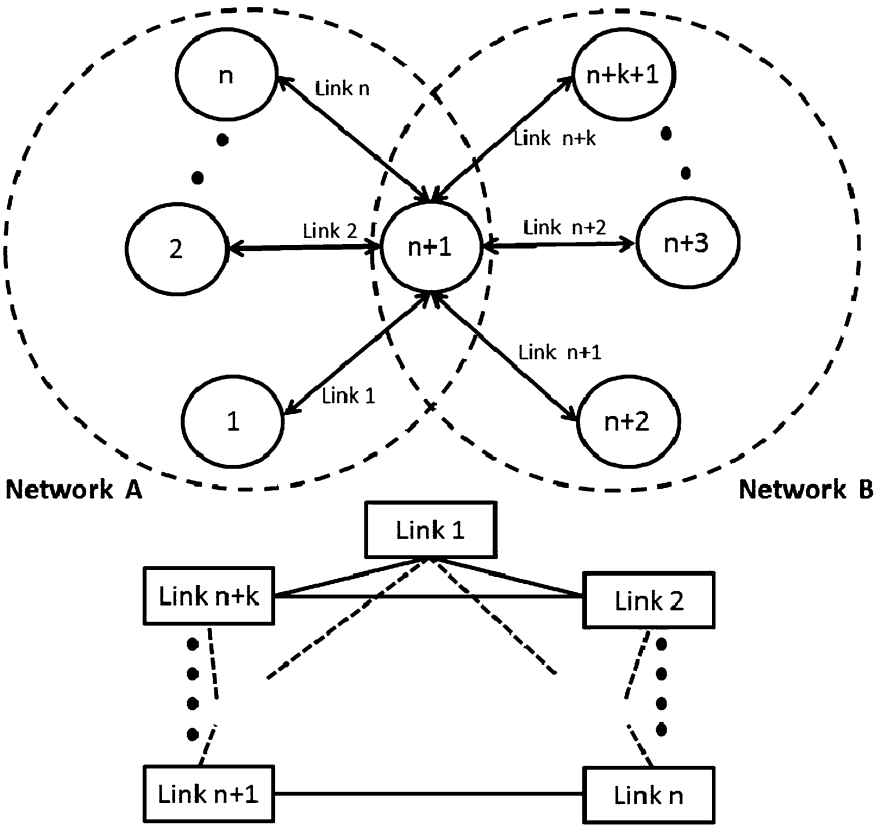

Fig. 1 An $(n + k)$-link network scenario and conflict graph

will be a collision. Let $(n + k)$ denote the total number of links in the network. In a typical CSMA network, the transmitter of node m backs off for a random period before it sends a packet to its destination node, if the channel is idle. If the channel is busy, the transmitter freezes its backoff counter until the channel is idle again. This backoff time, or the waiting time, for each link m is exponentially distributed with mean $1/R_m$. The objective in this chapter is to determine the optimal values of the mean transmission rates R_m, $m = 1, 2, \ldots, n + k$, so that the throughput in the network is maximized. For this purpose, a Markovian model is used with states defined as $x^i : \mathcal{A} \to \{0, 1\}^{n+k}$, where $i \in \mathcal{A}$ represents the status of the network, which takes the value of 1 for an active link and 0 represents an idle link. For example, if the mth link in state i is active, then $x^i_m = 1$.

Previous work assumes that the propagation delay between neighboring nodes is zero (cf. [5, 10]). Since propagation delays enable the potential for collisions, there exists motivation to maximize the throughput in the network in the presence of these delays. Additionally, collisions due to hidden terminals are possible, and this chapter captures the effect of hidden terminals in the CSMA Markov chain described in the following section.

3 CSMA Markov Chain

Formulations of Markov models for capturing the MAC layer dynamics in CSMA networks were developed in [5, 14]. The stationary distribution of the states and the balance equations were developed and used to quantify the throughput. Recently, a continuous-time CSMA Markov model without collisions was used in [10] to develop an adaptive CSMA to maximize throughput. Collisions were introduced in [9] in the Markov model, and the mean transmission length of the packets is used as the control variable to maximize the throughput. Since most applications experience random length of packets, the transmission rates (packets/unit time), R_m, $m = 1, 2, \ldots, n$, provide a practical measure.

The model for waiting times is based on the CSMA random access protocol. The probability density function of the waiting time T_m is given by

$$f_{T_m}(t_m) = \begin{cases} R_m \exp(-R_m t_m), & t_m \geq 0, \\ 0, & t_m < 0. \end{cases}$$

Due to the sensing delay experienced by the network nodes, the probability that link m becomes active within a time duration of δT_s from the instant link l becomes active is

$$p_{c_m} \triangleq 1 - \exp(-R_m \delta T_s) \tag{1}$$

by the memoryless property of the exponential random variable. Thus, the rate of transition G_i to one of the non-collision states in the Markov chain in Fig. 2 is defined as

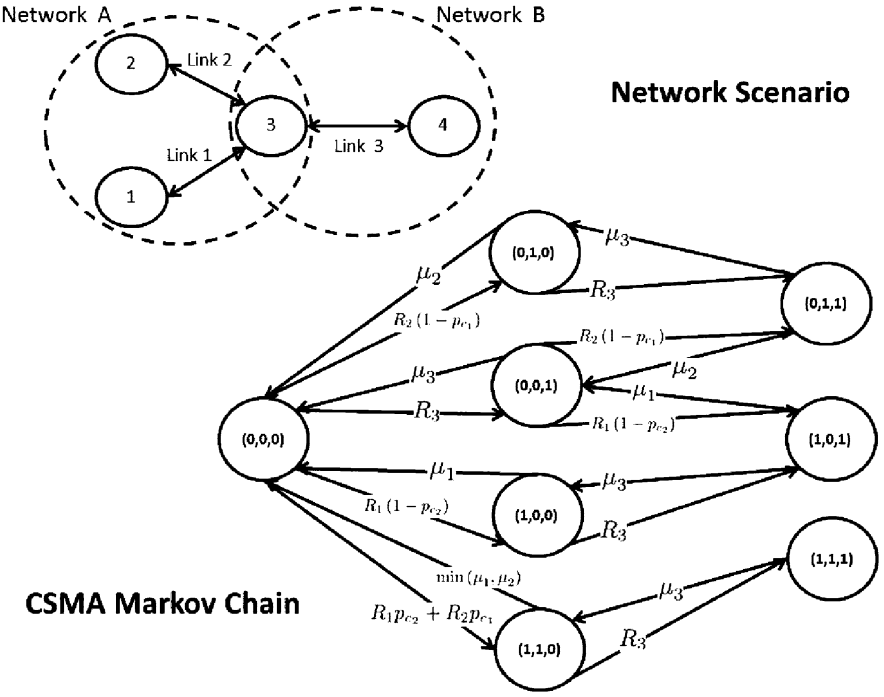

Fig. 2 CSMA Markov chain with collision states for a 3-link network scenario with hidden terminals

$$G_i = \sum_{u=1}^{n} \left(x_u^i R_u \prod_{l \neq u} \left(1 - p_{c_l}\right)^{\left(1-x_l^i\right)} \right). \tag{2}$$

The rate of transition G_i to one of the collision states is given by

$$G_i = \sum_{u=1}^{n} \left(x_u^i R_u \prod_{l \neq u} \left(p_{c_l}\right)^{x_l^i} \left(1 - p_{c_l}\right)^{\left(1-x_l^i\right)} \right). \tag{3}$$

For example, the state $(1, 1, 0)$ in Fig. 2 represents the collision state (for network A), which occurs when a link tries to transmit within a time span of δT_s from the instant another link starts transmitting.

The primary objective of modeling the network as a continuous CSMA Markov chain is that the probability of collision-free transmission needs to be maximized. For this purpose, the rate r_i is defined as

$$r_i \triangleq \begin{cases} \log\left\{\dfrac{\sum_{u=1}^{n}\left(x_u^i R_u \prod_{l\neq u}\left(1-p_{c_l}\right)^{\left(1-x_l^i\right)}\right)}{\sum_{u=1}^{n} x_u^i \mu_u}\right\}, & i \in \mathcal{A}_T \\ \dfrac{\sum_{u=1}^{n}\left(x_u^i R_u \prod_{l\neq u}\left(p_{c_l}\right)^{x_l^i}\left(1-p_{c_l}\right)^{\left(1-x_l^i\right)}\right)}{\min\limits_{m:x_m^i\neq 0}(\mu_m)}, & i \in \mathcal{A}_C \\ 1, & i \in \mathcal{A}_I, \end{cases} \tag{4}$$

so that the stationary distribution of the continuous-time Markov chain can be defined as

$$p(i) \triangleq \frac{\exp(r_i)}{\sum_j \exp\left(r_j\right)}, \tag{5}$$

where, in (4), $1/\mu_m$ is the mean transmission length of the packets if the network is in one of the states in set $\mathcal{A}_T$ in sensing region A. The set $\mathcal{A}_T \triangleq \mathcal{A}_C^c \setminus (0,0)^T$ represents the set of all collision-free transmission states, where the elements in the set $\mathcal{A}_C$ represent the collision states, and the elements in the set $\mathcal{A}_C^c$ represent the non-collision states. The set $\mathcal{A}_I$ represents the inactive state, i.e., $x^i = (0,0,0)$. In (4), the definitions for the rate of transitions in (2) and (3) are used, and (5) satisfies the detailed balance equation (cf. [11]).

In addition, if there are hidden terminals (HT) in the network as shown in Fig. 2, then r_i can be defined for the sensing region B in a similar way as defined for sensing region A in (4). Let sets $\mathcal{B}_T$, $\mathcal{B}_C$, and $\mathcal{B}_I$ represent the collision-free transmission states, collision states, and the inactive states, respectively. Based on the transmission, collision, and idle states of the links in the sensing regions A and B, i belongs to one of the combinations of the sets $\mathcal{A}_T$, $\mathcal{A}_C$, $\mathcal{A}_I$, $\mathcal{B}_T$, $\mathcal{B}_C$, and $\mathcal{B}_I$. Therefore (cf. [5]),

$$r_i \triangleq \begin{cases} F_A F_B, & i \in \mathcal{A}_T \cup \mathcal{B}_T \\ G_A F_B, & i \in \mathcal{A}_C \cup \mathcal{B}_T \\ F_B, & i \in \mathcal{A}_I \cup \mathcal{B}_T \\ F_A G_B, & i \in \mathcal{A}_T \cup \mathcal{B}_C \\ G_A G_B, & i \in \mathcal{A}_C \cup \mathcal{B}_C \\ G_B, & i \in \mathcal{A}_I \cup \mathcal{B}_C \\ F_A, & i \in \mathcal{A}_T \cup \mathcal{B}_I \\ G_A, & i \in \mathcal{A}_C \cup \mathcal{B}_I \\ 1, & i \in \mathcal{A}_I \cup \mathcal{B}_I, \end{cases}$$

where

$$F_A \triangleq \log\left\{\frac{\sum_{u=1}^{n}\left(x_u^i R_u \prod_{l\neq u}\left(1-p_{c_l}\right)^{\left(1-x_l^i\right)}\right)}{\sum_{u=1}^{n} x_u^i \mu_u}\right\},$$

$$G_A \triangleq \frac{\sum_{u=1}^{n}\left(x_u^i R_u \prod_{l\neq k}\left(p_{c_l}\right)^{x_l^i}\left(1-p_{c_l}\right)^{\left(1-x_l^i\right)}\right)}{\min\limits_{m:x_m^i\neq 0}\left(\mu_m\right)}.$$

F_B and G_B can be defined similarly for network B in Fig. 1.

4 Throughput Maximization

To quantify the throughput, a log-likelihood function is defined as the summation over all the collision-free transmission states as

$$F\left(R\right) \triangleq \sum_{i\in(\mathcal{A}_T\cup\mathcal{B}_I)\cup(\mathcal{A}_I\cup\mathcal{B}_T)} \log\left(p\left(i\right)\right). \tag{6}$$

By using the definition for $p\left(i\right)$ in (5), the log-likelihood function can be rewritten as

$$\begin{aligned}
F(R) = &\sum_{u=1}^{n}\log\left(\frac{R_u}{\mu_u}\right) - (n-1)\sum_{u=1}^{n} R_u\delta T_s \\
&+ \sum_{v=1+1}^{k+n}\log\left(\frac{R_v}{\mu_v}\right) - (k-1)\sum_{v=n+1}^{k+n} R_v\delta T_s \\
&- (n+k)\log\Bigg[\sum_{i\in\mathcal{A}_T\cup\mathcal{B}_T}\exp\left(F_A F_B\right) + \sum_{i\in\mathcal{A}_C\cup\mathcal{B}_T}\exp\left(G_A F_B\right) \\
&\quad + \sum_{i\in\mathcal{A}_I\cup\mathcal{B}_T}\exp\left(F_B\right) + \sum_{i\in\mathcal{A}_T\cup\mathcal{B}_C}\exp\left(F_A G_B\right) \\
&\quad + \sum_{i\in\mathcal{A}_C\cup\mathcal{B}_C}\exp\left(G_A G_B\right) + \sum_{i\in\mathcal{A}_I\cup\mathcal{B}_C}\exp\left(G_B\right) \\
&\quad + \sum_{i\in\mathcal{A}_T\cup\mathcal{B}_I}\exp\left(F_A\right) + \sum_{i\in\mathcal{A}_C\cup\mathcal{B}_I}\exp\left(G_A\right) + \sum_{i\in\mathcal{A}_I\cup\mathcal{B}_I}\exp\left(1\right)\Bigg].
\end{aligned} \tag{7}$$

The function $F\left(R\right)$ is convex (cf. [6]), and $F\left(R\right) \leq 0$ since $\log\left(p\left(x^i\right)\right) \leq 0$. The optimization problem is defined as

$$\min_{R} \left(-F\left(R\right)\right). \tag{8}$$

In addition to maximizing the log-likelihood function, certain constraints must be satisfied. The service rate $S\left(R\right)$ at each transmitter of a link needs to be equal to the arrival rate λ, and the chosen mean transmission rates $R_k, k = 1, 2, \ldots, n$, need to be nonnegative. Thus, the optimization problem can be formulated as

$$\min_{R} \left(-F\left(R\right)\right)$$

subject to

$$\log \lambda - \log S\left(R\right) = 0, \tag{9}$$

and

$$-R \leq 0, \tag{10}$$

where $R \in \mathbb{R}^n$, $S\left(R\right) \in \mathbb{R}^{n-1}$, and $\lambda \in \mathbb{R}^{n-1}$. The service rate for a link is the rate at which a packet is transmitted, and is quantified for sensing region A as

$$S_m(R) \triangleq \frac{\exp\left(\log\left(\frac{R_k \prod_{l \neq m} \exp(-R_l \delta T_s)}{\mu_m}\right)\right)}{\sum_j \exp\left(r_j\right)},$$

$m = 1, 2, \ldots, n-1$, and the denominator is defined in (4). Service rates for sensing region B can be defined similarly. Note that $\log \lambda_m - \log S_m\left(R\right) = 0$, and $\lambda_m > 0$ is convex for all m. The optimization problem defined above is a convex-constrained nonlinear programming problem, and obtaining an analytical solution is difficult. There are numerical techniques adopted in the literature which have investigated such problems in detail [1, 2, 6, 12]. As detailed in Sect. 5, a suitable numerical optimization algorithm is employed to solve the optimization problem defined in (8)–(10).

5 Simulation Results

The constrained convex nonlinear programming problem defined in (8)–(10) is solved by optimizing the mean transmission rates $R_m, m = 1, 2, \ldots, n + k$, of the transmitting nodes in the network of Fig. 1. A MATLAB built-in function `fmincon` is used to solve the optimization problem by configuring it to use the interior point algorithm (cf. [7, 8]).

Once the mean transmission rates are optimized, they are fixed in a simulation (developed in MATLAB) that uses the CSMA MAC protocol. The function `fmincon` solves the optimization problem only for a set of feasible arrival rates.

Table 1 Optimal values of the mean transmission rates for a 3-link collision network with hidden terminals (refer to Fig. 2) for various values of sensing delays. The optimum values of the mean transmission rates are the solution to the constrained problem defined in (8)–(10)

	Max. Feasible arrival rate			Opt. Mean TX rate		
Sensing delay	λ_1	λ_2	λ_3	R_1	R_2	R_3
0.001	0.2	0.2	0.1	3.94	3.94	1.96
0.01	0.18	0.17	0.11	3.78	3.58	2.23
0.1	0.12	0.12	0.1	2.56	2.56	1.65

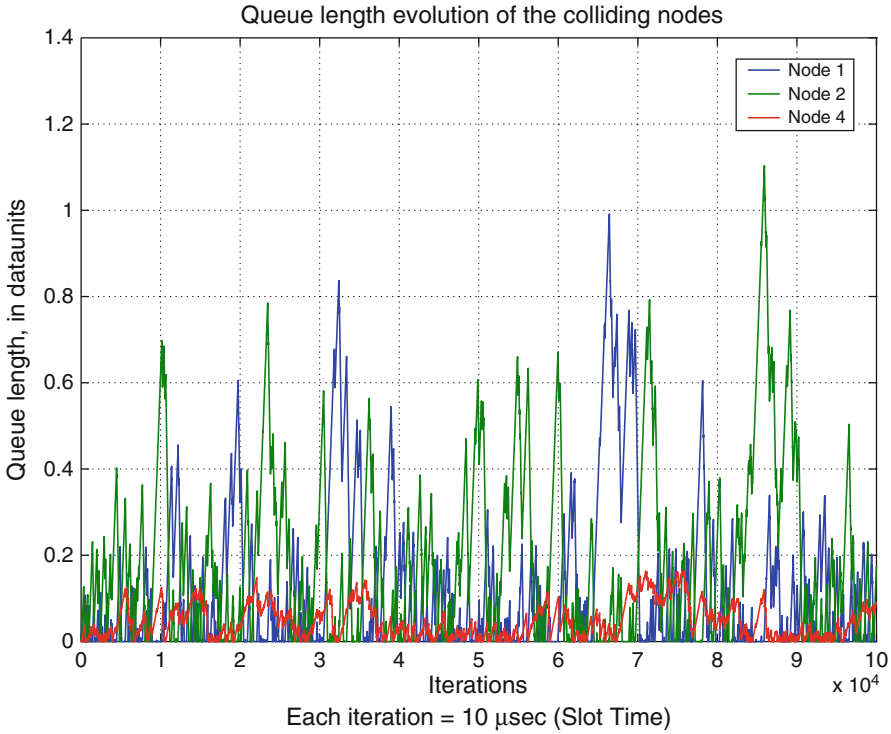

Fig. 3 Queue lengths of nodes 1, 2, and 4 transmitting to the same node 3. The optimum values of the mean transmission rates are the solution to the constrained problem defined in (8)–(10). All nodes are in the sensing region, and $\delta T_s = 0.01$ ms, $R_1 = 3.78$ dataunits/ms, $R_2 = 3.58$ dataunits/ms, $R_3 = 2.23$ dataunits/ms, $\lambda_1 = 0.02$ dataunits/ms, $\lambda_2 = 0.05$ dataunits/ms, $\lambda_3 = 0.05$ dataunits/ms

A slot time of 10 μs is used, and the mean transmission lengths of the packets, $1/\mu_m$, $m = 1, 2, \ldots, n + k$, are set to 1 ms. Further, a stable (and feasible) set of arrival rates, in the sense that the queue lengths at the transmitting nodes are stable, are chosen before the simulation.

The collision network of Fig. 1 is simulated using the platform explained above. The optimal values of the mean transmission rates, R_1, R_2, and R_3, are obtained and tabulated as shown in Table 1 for different values of the sensing delay δT_s (note that in the scenario of Fig. 1, the sensing delay applies to the nodes in network A). The capacity of the channel is normalized to 1 dataunit/ms. The mean transmission lengths of the packets are $1/\mu_1 = 1/\mu_2 = 1/\mu_3 = 1$ ms.

A simulation of a CSMA system with collisions is implemented in MATLAB. Figure 3 shows the evolution of the queue lengths of nodes 1, 2, and 4 (refer to Fig. 1) for a sensing delay of $\delta T_s = 0.01$ ms. The optimal mean transmission rates ($R_1 = 3.78$ dataunits/ms, $R_2 = 3.58$ dataunits/ms, $R_3 = 2.23$ dataunits/ms) are generated by `fmincon`, and the stable arrival rates of $\lambda_1 = 0.05$ dataunits/ms, $\lambda_2 = 0.05$ dataunits/ms, and $\lambda_3 = 0.01$ dataunits/ms are used.

6 Conclusion

A model for collisions caused due to both sensing delays and hidden terminals is developed and incorporated in the continuous CSMA Markov chain. A constrained optimization problem is defined, and a numerical solution is suggested. Simulation results are provided to demonstrate the stability of the queues for a given stable set of arrival rates. Future efforts will focus on including queue length constraints in the optimization problem and developing online solutions to the combined collision minimization and throughput maximization problem.

Acknowledgements This research is supported by a grant from AFRL Collaborative System Control STT.

References

1. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming—Theory and Algorithms (2nd edn.). Wiley, Hoboken (1993)
2. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1999)
3. Bianchi, G.: IEEE 802.11—Saturation throughput analysis. IEEE Commun. Lett. **2**(12), 318–320 (1998)
4. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Select. Commun. **18**(3), 535–547 (2000)
5. Boorstyn, R., Kershenbaum, A., Maglaris, B., Sahin, V.: Throughput analysis in multihop CSMA packet radio networks. IEEE Trans. Commun. **35**(3), 267–274 (1987)
6. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
7. Byrd, R.H., Gilbert, J.C.: A trust region method based on interior point techniques for nonlinear programming. Math. Progr. **89**, 149–185 (1996)
8. Byrd, R.H., Hribar, M.E., Jorge Nocedal, Z.: An interior point algorithm for large scale nonlinear programming. SIAM J. Optim. **9**, 877–900 (1999)

9. Jiang, L., Walrand, J.: Approaching throughput-optimality in distributed CSMA scheduling algorithms with collisions. IEEE/ACM Trans. Netw. **19**(3), 816–829 (2011)
10. Jiang, L., Walrand, J.: A distributed CSMA algorithm for throughput and utility maximization in wireless networks. IEEE/ACM Trans. Netw. **18**(3), 960–972 (2010)
11. Kelly, K.P.: Reversibility and Stochastic Networks. Wiley, Chichester (1979)
12. Luenberger, D.G.: Introduction to Linear and Nonlinear Programming. Addison-Wesley, Reading (1973)
13. Marbach, P., Eryilmaz, A., Ozdaglar, A.: Achievable rate region of CSMA schedulers in wireless networks with primary interference constraints. In: Proceedings of the IEEE Conference on Decision and Control, pp. 1156–1161 (2007)
14. Wang, X., Kar, K.: Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks. In: Proceedings of the IEEE Annual Joint Conference IEEE Computation and Communication Societies INFOCOM 2005, vol. 1, pp. 23–34 (2005)

Optimal Formation Switching with Collision Avoidance and Allowing Variable Agent Velocities

Dalila B.M.M. Fontes, Fernando A.C.C. Fontes, and Amélia C.D. Caldeira

Abstract We address the problem of dynamically switching the geometry of a formation of a number of undistinguishable agents. Given the current and the final desired geometries, there are several possible allocations between the initial and final positions of the agents as well as several combinations for each agent velocity. However, not all are of interest since collision avoidance is enforced. Collision avoidance is guaranteed through an appropriate choice of agent paths and agent velocities. Therefore, given the agent set of possible velocities and initial positions, we wish to find their final positions and traveling velocities such that agent trajectories are apart, by a specified value, at all times. Among all the possibilities we are interested in choosing the one that minimizes a predefined performance criteria, e.g. minimizes the maximum time required by all agents to reach the final geometry. We propose here a dynamic programming approach to solve optimally such problems.

Keywords Autonomous agents • Optimization • Dynamic programming • Agent formations • Formation geometry • Formation switching • Collision avoidance

D.B.M.M. Fontes (✉)
Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
e-mail: fontes@fep.up.pt

F.A.C.C. Fontes
Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
e-mail: faf@fe.up.pt

A.C.D. Caldeira
Departamento de Matemática, Instituto Superior de Engenharia do Porto, R. Dr. António Bernardino de Almeida 431, 4200-072 Porto, Portugal
e-mail: acd@isep.ipp.pt

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_11,

1 Introduction

In this paper, we study the problem of switching the geometry of a formation of undistinguishable agents by minimizing some performance criterion. The questions addressed are, given the initial positions and a set of final desirable positions, which agent should go to a specific final position, how to avoid collision between the agents, and which should be the traveling velocities of each agent between the initial and final positions. The performance criterion used in the example explored is to minimize the maximum traveling time, but the method developed—based on dynamic programming—is sufficiently general to accommodate many different criteria.

Formations of undistinguishable agents arise frequently both in nature and in mobile robotics. The specific problem of switching the geometry of a formation arises in many cooperative agents missions, due to the need to adapt to environmental changes or to adapt to new tasks. An example of the first type is when a formation has to go through a narrow passage, or deviate from obstacles, and must reconfigure to a new geometry. Examples of adaptation to new tasks arise in robot soccer teams: when a team is in an attack formation and loses the ball, it should switch to a defence formation more appropriate to the new task. Another example arises in the detection and containment of a chemical spillage, the geometry of the formation for the initial task of surveillance, should change after detection occurs, switching to a formation more appropriate to determine the perimeter of the spill.

Research in coordination and control of teams of several agents (that may be robots, ground, air, or underwater vehicles) has been growing fast in the past few years. Application areas include unmanned aerial vehicles (UAVs) [4, 18], autonomous underwater vehicles (AUVs) [16], automated highway systems (AHSs) [3, 17], and mobile robotics [20, 21]. While each of these application areas poses its own unique challenges, several common threads can be found. In most cases, the vehicles are coupled through the task they are trying to accomplish, but are otherwise dynamically decoupled, meaning the motion of one does not directly affect the others. For a survey in cooperative control of multiple vehicles systems, see, e.g., the work by Murray [11]. Regarding research on the optimal formation switching problem, it is not abundant, although it has been addressed by some authors. Desai et al. in [5], model mobile robots formation as a graph. The authors use the so-called "control graphs" to represent the possible solutions for formation switching. In this method, for a graph having n vertices there are $n!(n-1)!/2^{n-1}$ control graphs, and switching can only happen between predefined formations. The authors do not address collision or velocity issues. Hu and Sastry [9] study the problems of optimal collision avoidance and optimal formation switching for multiple agents on a Riemannian manifold. However, no choice of agent traveling velocity is considered. It is assumed that the underlying manifold admits a group of isometries, with respect to which the Lagrangian function is invariant. A reduction method is used to derive optimality conditions for the solutions. In [19] Yamagishi describes a decentralized controller for the reactive formation switching of a team of autonomous mobile robots. The focus is on how a structured formation of agents can

reorganize into a nonrigid formation based on changes in the environment. The controller utilizes nearest-neighbor artificial potentials (social rules) for collision-free formation maintenance and environmental changes act as a stimulus for switching between formations. A similar problem, where a set of agents must perform a fixed number of different tasks on a set of targets, has been addressed by several authors. The methods developed include exhaustive enumeration (see Rasmussen et al. [13]), branch-and-bound (see Rasmussen and Shima [12]), network models (see Schumacher et al. [14, 15]), and dynamic programming (see Jin et al. [10]). None of these works address velocity issues.

A problem of formation switching has also been addressed in [6, 7] using dynamic programming. However, the possible use of different velocities for each agent was not addressed. But the possibility of slowing down some of the agents might, as we will show in an example, achieve better solutions while avoiding collision between agents. We propose a dynamic programming approach to solve the problem of formation switching with collision avoidance and agent velocities selection, that is, the problem of deciding which agent moves to which place in the next formation guaranteeing that at any time the distance between any two of them is at least some predefined value. In addition, each agent can also explore the possibility of modifying its velocity to avoid collision, which is a main distinguishing feature from previous work. The formation switching performance is given by the time required for all agents to reach their new position, which is given by the maximum traveling time amongst individual agent traveling times. Since we want to minimize the time required for all agents to reach their new position, we have to solve a minmax problem. However, the methodology we propose can be used with any separable performance function. The problem addressed here should be seen as a component of a framework for multiagent coordination, incorporating also the trajectory control component [8], which allows to maintain or change formation while following a specified path in order to perform cooperative tasks.

This paper is organized as follows. In the next section, the problem of optimal reorganization of agent formations with collision avoidance is described and formally defined. In Sect. 3, a dynamic programming formulation of the problem is given and discussed. In Sect. 4, we discuss computational implementation issues of the dynamic programming algorithm, namely an efficient implementation of the main recursion as well as efficient data representations. A detailed description of the algorithms is also provided. Next, an example is reported to show the solution modifications when using velocities selection and collision avoidance. Some conclusions are drawn in the final section.

2 The Problem

In our problem a team of N identical agents has to switch from their current formation to some other formation (i.e., agents have a specific goal configuration not related to the positions of the others), possibly unstructured, with collision avoidance. To address collision avoidance, we impose that the trajectories of the

agents must satisfy the separation constraint that at any time the distance between (the center of) any two of them is at least ϵ, for some positive ϵ. (So, ϵ should be at least the diameter of an agent.) The optimal (joint) trajectories are the ones that minimize the maximum trajectory time of individual agents.

Our approach can be used either centralized or decentralized, depending on the agent capabilities. In the latter case, all the agents would have to run the algorithm, which outputs an optimal solution, always the same if many exist, since the proposed method is deterministic.

Regarding the new formation, it can be either a pre-specified formation or a formation to be defined according to the information collected by the agents. In both cases, we do a preprocessing analysis that allows us to come up with the desired locations for the next formation.

This problem can be restated as the problem of allocating to each new position exactly one of the agents, located in the old positions, and determine each agent velocity. From all the possible solutions we are only interested in the ones where agent collision is prevented. Among these, we want to find one that minimizes the time required for all agents to move to the target positions, that is, an allocation which has the least maximum individual agent traveling time.

To formally define the problem, consider a set of N agents moving in a space $\mathbb{R}^d$, so that at time t, agent i has position $q_i(t)$ in $\mathbb{R}^d$ (we will refer to $q_i(t) = (x_i(t), y_i(t))$ when our space is the plane $\mathbb{R}^2$). The position of all agents is defined by the N-tuple $Q(t) = [q_i(t)]_{i=1}^N$ in $\mathbb{R}^{d\times N}$. We assume that each agent is holonomic and that we are able to choose its velocity, so that its kinematic model is a simple integrator

$$\dot{q}_i(t) = \vartheta_i(t) \quad a.e.\ t \in \mathbb{R}^+.$$

The initial positions at time $t = 0$ are known and given by $A = [a_i]_{i=1}^N = Q(0)$. Suppose a set of M (with $M \geq N$) final positions in $\mathbb{R}^d$ is specified as $F = \{f_1, f_2, \ldots, f_M\}$.

The problem is to find an assignment between the N agents and N final positions in F. That is, we want to find an N-tuple $B = [b_i]_{i=1}^N$ of different elements of F, such that at some time $T > 0$, $Q(T) = B$ and all $b_i \in F$, with $b_i \neq b_k$. There are $\binom{M}{N} \cdot N!$ such N-tuples (the permutations of a set of N elements chosen from a set of M elements) and we want to find a procedure to choose an N-tuple minimizing a certain criterion that is more efficient than total enumeration.

The criterion to be minimized can be very general since the procedure developed is based on dynamic programming which is able to deal with general cost functions. Examples can be minimizing the total distance traveled by the agents

$$\text{Minimize} \sum_{i=1}^{N} \|b_i - a_i\|,$$

the total traveling time

$$\text{Minimize} \sum_{i=1}^{N} \|b_i - a_i\|/\|\vartheta_i\|,$$

or the maximum traveling time

$$\text{Minimize} \max_{i=1,\ldots,N} \|b_i - a_i\|/\|\vartheta_i\|.$$

We are also interested in selecting the traveling velocities of each agent. Assuming constant velocities, these are given by

$$\vartheta_i(t) = \vartheta_i = v_i \frac{b_i - a_i}{\|b_i - a_i\|},$$

where the constant speeds are selected from a discrete set $\Upsilon = \{V_{\min}, \ldots, V_{\max}\}$.

Moreover, we are also interested in avoiding collision between agents. We say that two agents i, k (with $i \neq k$) do not collide if their trajectories maintain a certain distance apart, at least ϵ, at all times. The non-collision conditions is

$$\|q_i(t) - q_k(t)\| \geq \epsilon \qquad \forall t \in [0, T], \tag{1}$$

where the trajectory is given by

$$q_i(t) = a_i + \vartheta_i(t)t, \quad t \in [0, T].$$

We can then define a logic-valued function c as

$$c(a_i, b_i, v_i, a_k, b_k, v_k) = \begin{cases} 1 \text{ if collision between } i \text{ and } k \text{ ocurs} \\ 0 \text{ otherwise} \end{cases}$$

With these considerations, the problem (in the case of minimizing the maximum traveling time) can be formulated as follows:

$$\min_{b_1,\ldots,b_N,v_1,\ldots,v_N} \max_{i=1,\ldots,N} \|b_i - a_i\|/v_i,$$

Subject to

$$\begin{aligned} & b_i \in F && \forall i, \\ & b_i \neq b_k && \forall i, k \text{ with } i \neq k, \\ & v_i \in \Upsilon, && \forall i, \\ & c(a_i, b_i, v_i, a_k, b_k, v_k) = 0, && \forall i, k \text{ with } i \neq k. \end{aligned}$$

Instead of using the set F of d-tuples, we can define a set $J = \{1, 2, \ldots, M\}$ of indexes to such d-tuples, and also a set $I = \{1, 2, \ldots, M\}$ of indexes to the agents. Let j_i in J be the target position for agent i, that is, $b_i = f_{j_i}$. Define also the distances $d_{ij} = \|f_j - a_i\|$ which can be pre-computed for all $i \in I$ and $j \in J$. Redefining, without changing the notation, the function c to take as arguments the indexes to the agent positions instead of the positions (i.e., $c(a_i, f_{j_i}, v_i, a_k, f_{j_k}, v_k)$ is simply represented as $c(i, j_i, v_i, k, j_k, v_k)$), the problem can be reformulated into the form

$$\min_{j_1,\ldots,j_N,v_1,\ldots,v_N} \max_{i=1,\ldots,N} d_{ij}/v_i,$$

Subject to

$$\begin{aligned} & j_i \in J && \forall i \in I, \\ & j_i \neq j_k && \forall i,k \in I \text{ with } i \neq k, \\ & v_i \in \Upsilon, && \forall i \in I, \\ & c(i, j_i, v_i, a_k, j_k, v_k) = 0, && \forall i,k \text{ with } i \neq k. \end{aligned}$$

3 Dynamic Programming Formulation

Dynamic programming (DP) is an effective method to solve combinatorial problems of a sequential nature. It provides a framework for decomposing an optimization problem into a nested family of subproblems. This nested structure suggests a recursive approach for solving the original problem using the solution to some subproblems. The recursion expresses an intuitive *principle of optimality* [2] for sequential decision processes; that is, once we have reached a particular state, a necessary condition for optimality is that the remaining decisions must be chosen optimally with respect to that state.

3.1 Derivation of the Dynamic Programming Recursion: The Simplest Problem

We start by deriving a DP formulation for a simplified version of problem: where collision is not considered and different velocities are not selected. The collision avoidance and the selection of velocities for each agent are introduced later.

Consider that there are N agents $i = 1, 2, \ldots, N$ to be relocated from known initial location coordinates to target locations indexed by set J. We want to allocate exactly one of the agents to each position in the new formation. In our model a stage i contains all states S such that $|S| \geq i$, meaning that i agents have been allocated to the targets in S. The DP model has N stages, with a transition occurring from a stage $i-1$ to a stage i, when a decision is made about the allocation of agent i.

Define $f(i, S)$ to be the value of the best allocation of agents $1, 2, \ldots, i$ to the i targets in set S, that is, the allocation requiring the least maximum time the agents take to go to their new positions. Such value is found by determining the least maximum agent traveling time between its current position and its target position. For each agent, i, the traveling time to the target position j is given by d_{ij}/v_i. By the previous definition, the minimum traveling time of the $i-1$ agents to the target positions in set $S \setminus \{j\}$ is given by $f\ (i-1, S \setminus \{j\})$. From the above, the minimum traveling time of all i agents to the target positions in S they are assigned to, given that agent i travels at velocity v_i, without agent collisions, is obtained by examining all possible target locations $j \in S$ (see Fig. 1).

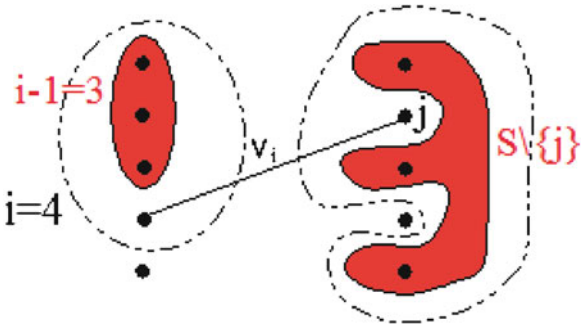

Fig. 1 Dynamic programming recursion for an example with $N = 5$ and stage $i = 4$

The *dynamic programming recursion* is then defined as

$$f(i, S) = \min_{j \in S} \left\{ d_{ij}/v_i \vee f(i-1, S \setminus \{j\}) \right\}, \tag{2}$$

where $X \vee Y$ denotes the maximum between X and Y.

The *initial conditions* for the above recursion are provided by

$$f(1, S) = \min_{j \in S} \left\{ d_{1j}/v_1 \right\}, \quad \forall S \subseteq J, \tag{3}$$

and all other states are initialized as not yet computed.

Hence, the optimal value for the performance measure, that is, the minimum traveling time needed for all N agents to assume their new positions in J, is given by

$$f(N, J). \tag{4}$$

3.2 Considering Collision Avoidance and Velocities Selection

Recall function c for which $c(i, j, v_i, a, b, v_a)$ takes value 1 if there is collision between pair of agents i and a traveling to positions j and b with velocities v_i and v_a, respectively, and takes value 0 otherwise. To analyze if the agent traveling through a newly defined trajectory collides with any agent traveling through previously determined trajectories, we define a recursive function. This function checks the satisfaction of the collision condition, given by (1), in turn, between the agent which had the trajectory defined last and each of the agents for which trajectory decisions have already been made. We note that by trajectory we understand not only the path between the initial and final positions but also a timing law and an implicitly defined velocity.

Consider that we are in state (i, S) and that we are assigning agent i to target j. Further let v_{i-1} be the traveling velocity for agent $i-1$. Since we are solving state (i, S) we need state $(i-1, S \setminus \{j\})$, which has already been computed. (If this is not

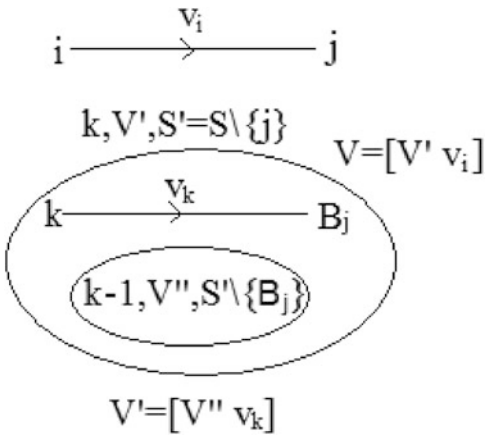

Fig. 2 Collision recursion

the case, then we must compute it first.) In order to find out if this new assignment is possible, we need to check if at any point in time agent i, traveling with velocity v_i will collide with any of the agents $1, 2, \ldots, i-1$ for which we have already determined the target assignment and traveling velocities.

Let us define a recursive function $\mathcal{C}(i, v_i, j, k, V, S)$ that assumes the value one if a collision occurs between agent i traveling with velocity v_i to j and any of the agents $1, 2, \ldots, k$, with $k < i$, traveling to their targets, in set S, with their respective velocities $V = [v_1 v_2 \cdots v_k]$ and assumes the value zero if no such collisions occurs. This function works in the following way (see Fig. 2):

1. First it verifies $c(i, v_i, j, k, v_k, \mathcal{B}_j)$, that is, it verifies if there is collision between trajectory $i \longrightarrow j$ at velocity v_i and trajectory $k \longrightarrow \mathcal{B}_j$ at velocity v_k, where $\mathcal{B}_j$ is the optimal target for agent k when targets in set $S \setminus \{j\}$ are available for agents $1, 2, \ldots, k$. If this is the case it returns the value 1.
2. Otherwise, if they do not collide, it verifies if trajectory $i \longrightarrow j$ at velocity v_i collides with any of the remaining agents. That is, it calls the collision function $\mathcal{C}(i, v_i, j, k-1, V', S')$, where $S' = S \setminus \{\mathcal{B}_j\}$ and $V = [V' \, v_k]$.

The collision recursion is therefore written as

$$\mathcal{C}(i, v_i, j, k, V, S) = \left\{c(i, v_i, j, k, v_k, \mathcal{B}_j) \vee \mathcal{C}(i, v_i, j, k-1, V', S')\right\} \tag{5}$$

where $\mathcal{B}_j = Best_j(k, V', S')$, $V = [V' \, v_k]$, $S' = S \setminus \{j\}$

The initial conditions for recursion (5) are provided by

$$\mathcal{C}(i, v_i, j, 1, v_1, \{k\}) = \{c(i, v_i, j, 1, v_1, k)\},$$

$\forall i \in I; \forall j, k \in J$ with $j \neq k; \forall v_i, v_1 \in \Upsilon$. All other states are initialized as not yet computed.

The dynamic programming recursion for the minimal time-switching problem with collision avoidance and velocities selection is then

$$f(i,V,S)=\min_{j\in S}\left\{d(i,j)/v_i \vee f(i-1,V',S') \vee M\cdot\mathcal{C}(i,v_i,j,i-1,V',S')\right\}, \quad (6)$$

where $V=[V'v_i]$, $S'=S\setminus\{j\}$, and $\mathcal{C}$ is the collision function.

The initial conditions are given by

$$f(1,v_1,\{j\})=\{d(1,j)/v_1\}, \forall j\in J \text{ and } \forall v_1\in\Upsilon. \quad (7)$$

All other states being initialized as not computed.

To determine the optimal value for our problem we have to compute

$$\min_{\text{all N-tuples V}} f(N,V,J).$$

4 Computational Implementation

The DP procedure we have implemented exploits the recursive nature of the DP formulation by using a backward–forward procedure. Although a pure forward DP algorithm can be easily derived from the DP recursion, (6) and (7), such implementation would result in considerable waste of computational effort since, generally, complete computation of the state space is not required. Furthermore, since the computation of a state requires information contained in other states, rapid access to state information should be sought.

The main advantage of the backward–forward procedure implemented is that the exploration of the state space graph, that is, the solution space, is based upon the part of the graph which has already been explored. Thus, states which are not feasible for the problem are not computed, since only states which are needed for the computation of a solution are considered. The algorithm is dynamic as it detects the needs of the particular problem and behaves accordingly.

States at stage 1 are either nonexistent or initialized as given in (3). The DP recursion, (2), is then implemented in a backward–forward recursive way. It starts from the final states (N,V,J) and while moving backward visits, without computing, possible states until a state already computed is reached. Initially, only states in stage 1, initialized by (3), are already computed. Then, the procedure is performed in reverse order, that is, starting from the state last identified in the backward process, it goes forward through computed states until a state (i,V',S') is found which has not yet been computed. At this point, again it goes backward until a computed state is reached. This procedure is repeated until the final states (N,V,J) for all V are reached with a value that cannot be improved by any other alternative solution. From these we choose the minimum one. The main advantage of this backward–forward recursive algorithm is that only intermediate states needed are visited and from these only the feasible ones that may yield a better solution are computed.

As said before, due to the recursive nature of (2), state computation implies frequent access to other states. Recall that a state is represented by a number, a sequence, and a set. Therefore, sequence operations like adding or removing an element and set operations like searching, deletion, and insertion of a set element must be performed efficiently.

4.1 Sequence Representation and Operation

Consider a sequence of length n, or an n-tuple, with k possible values for each element. (In the sequence of our example $n = N$ is the number of agents and $k = |\Upsilon|$ the number of possible velocity values.) There are k^n possible sequences to be represented. If sequences are represented by integers in the range $0 \sim k^n - 1$ then it is easy to implement sequence operations such as partitions. Thus, we represent a sequence as a numeral with n digits in the base k. The partition of a sequence with l digits that we are interested on is the one corresponding to the first $l - 1$ digits and the last digit. Such a partition can be obtained by performing the integer division in the base k and taking the remainder of such division.

Example 1. Consider a sequence of length $n = 4$ with $k = 3$ possible values v_0, v_1, and v_2. This is represented by numeral with n digits in the base k as

$$[v_1, v_0, v_2, v_1] \text{ is represented by} 1\ 0\ 2\ 1_3 = 1 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 1 \cdot 3^0 = 34$$

Partition of this sequence by the last element can be performed by integer division (DIV) in the base k and taking the remainder (MOD) of such division,

$$V = 1\ 0\ 2\ 1_3 = 34 \text{ can be split into } [V'\ v_i] \text{ as follows:}$$
$$V' = 1\ 0\ 2_3 = 1 \cdot 3^2 + 2 \cdot 3^0 = 11 = 34 \text{ DIV } 3$$

and

$$v_i = 1_3 = 1 = 34 \text{ MOD } 3.$$

4.2 Set Representation and Operation

A computationally efficient way of storing and operating sets is the bit-vector representation, also called the boolean array, whereby a *computer word* is used to keep the information related to the elements of the set. In this representation a universal set $U = \{1, 2, \ldots, n\}$ is considered. Any subset of U can be represented by a binary string (a computer word) of length n in which the ith bit is set to 1 if i is an element of the set, and set to 0 otherwise. So, there is a one-to-one correspondence between all possible subsets of U (in total 2^n) and all binary strings

Algorithm 1 DP for finding agent–target allocations and corresponding velocities

Input: The agent set, locations and velocities, the target set and locations, and the distance function;

Compute the distance for every pair agent-target (d_{ij});

Label all states as not yet computed;
$f(n, V, S) = \infty$;
for all $n = 1, 2, \ldots, N$, all V with n components, $S \in J$;

Initialize states at stage one as

$$f(1, V, \{j\}) = \{d_{1j}/v_1\}, \quad \forall V \in \Upsilon, j \in J.$$

Call $Compute(N, V, J)$ for all sequences V with N components;
Output: Solution performance;

Call $Allocation(N, V*, J)$;
Output: Agent targets and velocities;

of length n. Since there is also a one-to-one correspondence between binary strings and integers, the sets can be efficiently stored and worked out simply as integer numbers. A major advantage of such implementation is that the set operations, *location*, *insertion*, or *deletion* of a set element can be performed by directly addressing the appropriate bit. For a detailed discussion of this representation of sets see, e.g., the book by Aho et al. [1].

Example 2. Consider the Universal set U=$\{1, 2, 3, 4\}$ of $n = 4$ elements. This set and any of its subsets can be represented by a binary string of length 4, or equivalently its representation as an integer in the range 0–15.

$$U = \{1, 2, 3, 4\} \text{ is represented by } 1111_B = 15.$$

$$\text{A subset } A = \{1, 3\} \text{ is represented by } 0101_B = 5.$$

The flow of the algorithm is managed by Algorithm 1, which starts by labeling all states (subproblems) as not yet computed, that is, it assigns to them a ∞ value. Then, it initializes states in stage 1, that is subproblems involving 1 agent, as given by (3). After that, it calls Algorithm 2 with parameters (N, V, J). Algorithm 2, that implements recursion (2), calls Algorithm 3 to check for collisions every time it attempts to define one more agent–target allocation. This algorithm is used to find out whether the newly established allocation satisfies the collision regarding all previously defined allocations or not, feeding the result back to Algorithm 2. Algorithm 1, called after Algorithm 2 has finished,also implements a recursive function with which the solution structure, that is, the agent–target allocation, is retrieved.

Algorithm 2 Recursive function: compute optimal performance

```
Recursive Compute(i, V, S);
if f(i, V, S) ≠ ∞ then
  | return f(i, V, S) to caller;
end
Set min = ∞;
for each j ∈ S' do
  | S' = S \ {j}; V' = V DIV nvel; v_i = V MOD nvel;
  | Call Collision(i, v_i, j, i − 1, V', S')
  | if Col(i, j, i − 1, S') = 0 then
  |   | Call Compute(i − 1, V', S');
  |   | t_ij = d_ij / v_i;
  |   | aux = max(f(i − 1, V', S'), t_ij);
  |   | if aux ≤ min then
  |   |   | min = aux; best_j = j;
  |   | end
  | end
end
Store information: target B_j(i, V, S) = best_j; value f(i, V, S) = min;
Return: f(i, V, S);
```

Algorithm 2 is a recursive algorithm that computes the optimal solution cost, that is, it implements (2). This function receives three arguments: the agents to be allocated, their respective velocity values, and the set of target locations available to them, all represented by integer numbers. It starts by checking whether the specific state (i, V, S) has already been computed or not. If so, the program returns to the point where the function was called; otherwise, the state is computed. To compute state (i, V, S), all possible target locations $j \in S$ that might lead to a better subproblem solution are identified. The function is then called with arguments $(i-1, V', S')$, where $V' = V\ DIV\ nvel$ (V' is the subsequence of v containing the first $i-1$ elements, and $nvel$ the number of possible velocity values) and $S' = S \setminus \{j\}$, for every j such that allocating agent i to target j does not lead to any collision with previously defined allocations. This condition is verified by Algorithm 3.

Algorithm 3 is a recursive algorithm that checks the collision of a specific agent-target allocation traveling at a specific velocity with the set of allocations and velocities previously established, that is, it implements (5). This function receives six arguments: the newly defined agent–target allocation $i \longrightarrow j$ and its traveling velocity v_i and the previously defined allocations and respective velocities to check with, that is agents $1, 2, \ldots, k$, their velocities and their target locations S. It starts by checking the collision condition, given by (1), for the allocation pair $i \longrightarrow j$ traveling at velocity v_i and $k \longrightarrow \mathcal{B}_j$ traveling at velocity v_k, where $\mathcal{B}_j$ is the optimal target for agent k when agents $1, 2, \ldots, k$ are allocated to targets in S. If there is collision it returns 1; otherwise it calls itself with arguments $(i, v_i, j, k-1, V', S \setminus \{\mathcal{B}_j\})$.

Algorithm 4 is also a recursive algorithm and it backtracks through the information stored while solving subproblems, in order to retrieve the solution structure, that is, the actual agent–target allocation and agent velocity. This algorithm works

Algorithm 3 Recursive function: find if the trajectory of the allocation $i \rightarrow j$ at velocity v_i collides with any of the existing allocations to the targets in S at the specified velocities in V

Recursive *Collision*(i, v_i, j, k, V, S);

if $Col\,(i, v_i, j, k, V, S) \neq \infty$ **then**
  return $Col\,(i, v_i, j, k, V, S)$ to caller;
end
$B_j = \mathcal{B}_j(k, V, S)$;
if *collision condition is not satisfied* **then**
  $Col\,(i, v_i, j, k, V, S) = 1$;
  return $Col\,(i, v_i, j, k, V, S)$ to caller;
end
$S' = S \setminus \{B_j\}$;
$V' = V\ DIV\ nvel$;
$v_k = V\ MOD\ nvel$;
Call *Collision*$(i, v_i, j, k-1, V', S')$;

Store information: $Col\,(i, v_i, j, k, V, S) = 0$;

Return: $Col\,(i, v_i, j, k, V, S)$;

Algorithm 4 Recursive function: retrieve agent–target allocation and agents velocity

Recursive *Allocation*(i, V, S);

if $S \neq \varnothing$ **then**
  $v_i = VMODmvel$;
  $j =$*target*$\mathcal{B}_j(i, V, S)$;
  $Vloc(i) = v_i$;
  $Alloc(i) = j$;
  $V' = VDIVnvel$;
  $S' = S \setminus \{j\}$;
  CALL *Allocation*$(i-1, V', S')$;
end

Return: *Alloc*;

backward from the final state (N, V^*, J), corresponding to the optimal solution obtained, and finds the partition by looking at the agent traveling velocity $v_N = V^*\ MOD\ nvel$ and at the target stored for this state $\mathcal{B}_j(N, V^*, J)$, with which it can build the structure of the solution found. Algorithm 3 receives three arguments: the agents, their traveling velocity, and the set of target locations. It starts by checking whether the agent current locations set is empty. If so, the program returns to the point where the function was called; otherwise the backtrack information of the state is retrieved and the other needed states evaluated.

5 An Example

An example is given to show how agent–target allocations are influenced by imposing that no collisions are allowed both with a single fixed velocity value for all agents and with the choice of agent velocities from three different possible values. In this example we have decided to use d_{ij} as the Euclidian distance although any other distance measure may have been used.

The separation constraints impose, at any point in time, the distance between any two agent trajectories to be at least 15 points; otherwise it is considered that those two agents collide.

Consider four agents, A, B, C, and D with random initial positions as given in Table 1 and four target positions 1, 2, 3, and 4 in a diamond formation as given in Table 2. We also consider three velocity values: $v_1 = 10$, $v_2 = 30$, $v_3 = 50$.

In Fig. 3 we give the graphical representation of the optimal agent–target allocation found, when a single velocity value is considered and collisions are allowed and no collisions are allowed, respectively.

As it can be seen in the top part of Fig. 3, that is, when collisions are allowed, the trajectory of agents A and D do not remain apart, by 15 points, at all times. Therefore, when no collisions are enforced the agent–target allocation changes with an increase in the time that it takes for all agents to assume their new positions.

In Fig. 4 we give the graphical representation of an optimal agent–target allocation found, when there are three possible velocity values to choose from and collisions are allowed and no collisions are allowed, respectively.

As it can be seen in the top part of the Fig. 4, that is, when collisions are allowed, the trajectory of agents A and D do not remain apart, by 15 points, at all times, since the agents move at the same velocity. Therefore, when no collisions are enforced although the agent–target allocation remains the same, agent A has it velocity decreased and therefore its trajectory no longer collides with the trajectory of agent D. Furthermore, since agents A trajectory is smaller this can be done with no increase in the time that it takes for all agents to assume their new positions.

Table 1 Agents random initial location

	Location	
	x_i	y_i
Agent A	35	185
Agent B	183	64
Agent C	348	349
Agent D	30	200

Table 2 Target locations, in diamond formation

	Location	
	x_i	y_i
Target 1	95	258
Target 2	294	258
Target 3	195	169
Target 4	195	347

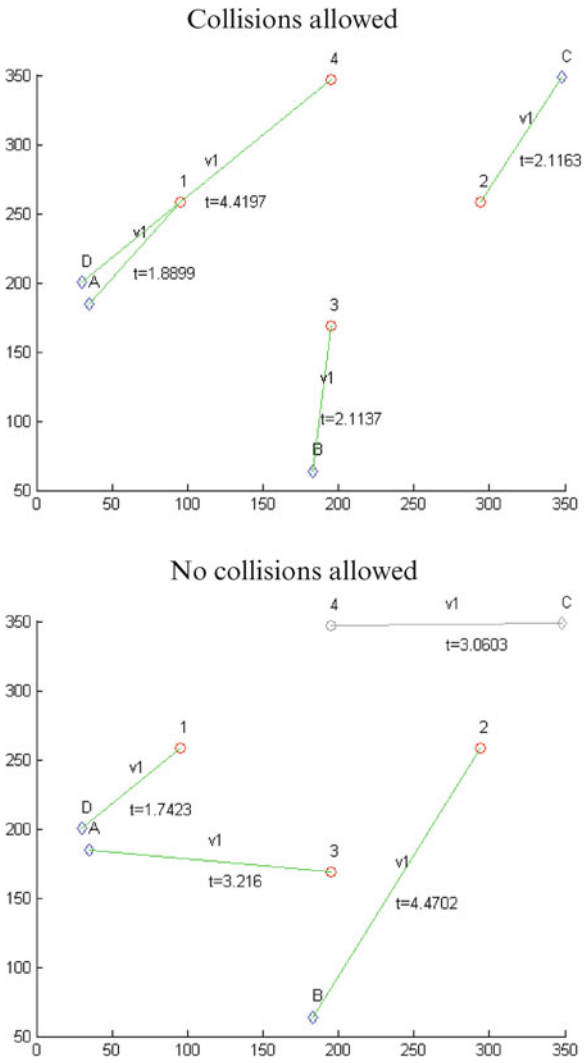

Fig. 3 Comparison of solutions with and without collision for the single velocity case

6 Conclusion

We have developed an optimization algorithm to decide how to reorganize a formation of vehicles into another formation of different shape with collision avoidance and agent traveling velocity choice, which is a relevant problem in cooperative control applications. The method proposed here should be seen as a component of a framework for multiagent coordination/cooperation, which must necessarily include other components such as a trajectory control component.

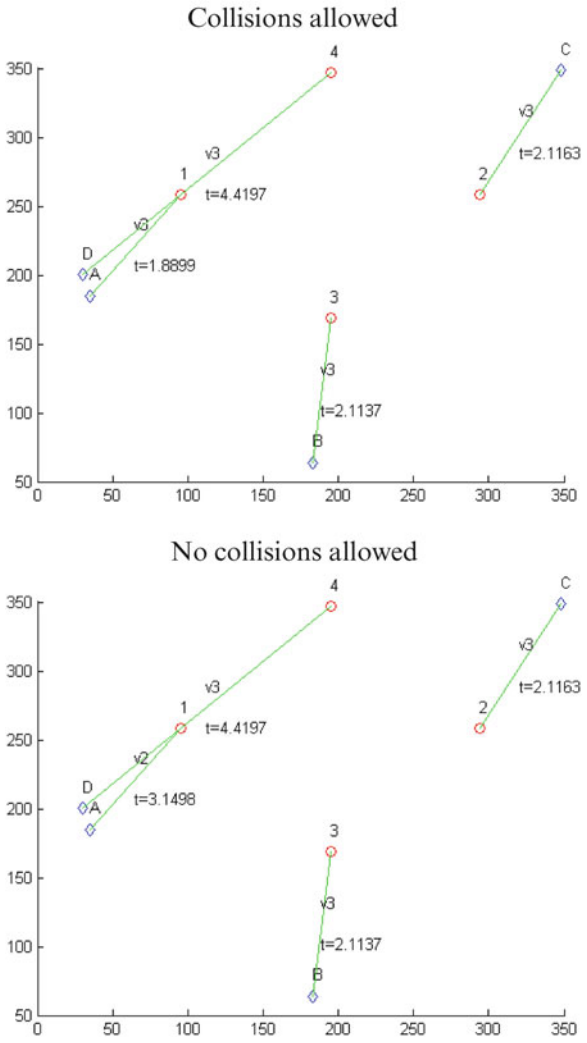

Fig. 4 Comparison of the solutions with and without collision for the velocity choice case

The algorithm proposed is based on a dynamic programming approach that is very efficient for small dimensional problems. As explained before, the original problem is solved by combining, in an efficient way, the solution to some subproblems. The method efficiency improves with the number of times the subproblems are reused, which obviously increases with the number of feasible solutions.

Moreover, the proposed methodology is very flexible, in the sense that it easily allows for the inclusion of additional problem features, e.g., imposing geometric constraints on each agent or on the formation as a whole, using nonlinear trajectories, among others.

Acknowledgements Research supported by COMPETE & FEDER through FCT Projects PTDC/EEA-CRO/100692/2008 and PTDC/EEA-CRO/116014/2009.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: Data Structures and Algorithms. Addison-Wesley, Reading MA (1983)
2. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton, USA (1957)
3. Bender, J.G.: An overview of systems studies of automated highway systems. IEEE Trans. Vehicular Tech. **40**(1 Part 2), 82–99 (1991)
4. Buzogany, L.E., Pachter, M., d'Azzo, J.J.: Automated control of aircraft in formation flight. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, 1349–1370, Monterey, California (1993)
5. Desai, J.P., Ostrowski, P., Kumar, V.: Modeling and control of formations of nonholonomic mobile robots. IEEE Trans. Robotic Autom. **17**(6), 905–908 (2001)
6. Fontes, D.B.M.M., Fontes, F.A.C.C.: Optimal reorganization of agent formations. WSEAS Trans. Syst. Control. **3**(9), 789–798 (2008)
7. Fontes, D.B.M.M., Fontes, F.A.C.C.: Minimal switching time of agent formations with collision avoidance. Springer Optim. Appl. **40**, 305–321 (2010)
8. Fontes, F.A.C.C., Fontes, D.B.M.M., Caldeira, A.C.D.: Model predictive control of vehicle formations. In: Pardalos, P., Hirsch, M.J., Commander, C.W., Murphey, R. (eds.) Optimization and Cooperative Control Strategies. Lecture Notes in Control and Information Sciences, Vol. 381. Springer Verlag, Berlin (2009). ISBN: 978-3-540-88062-2
9. Hu, J., Sastry, S.: Optimal collision avoidance and formation switching on Riemannian manifolds. In: IEEE Conference on Decision and Control, IEEE 1998, vol. 2, 1071–1076 (2001)
10. Jin, Z., Shima, T., Schumacher, C.J.: Optimal scheduling for refueling multiple autonomous aerial vehicles. IEEE Trans. Robotic. **22**(4), 682–693 (2006)
11. Murray, R.M.: Recent research in cooperative control multivehicle systems. J. Dyn. Syst. Meas. Control. **129**, 571–583 (2007)
12. Rasmussen, S.J., Shima, T.: Branch and bound tree search for assigning cooperating UVAs to multiple tasks. In: Institute of Electrical and Electronic Engineers, American Control Conference 2006, Minneapolis, Minnesota, USA (2006)
13. Rasmussen, S.J., Shima, T., Mitchell, J.W., Sparks, A., Chandler, P.R.: State-space search for improved autonomous UAVs assignment algorithm. In: IEEE Conference on Decision and Control, Paradise Island, Bahamas (2004)
14. Schumacher, C.J., Chandler, P.R., Rasmussen, S.J.: Task allocation for wide area search munitions via iterative network flow. In: American Institute of Aeronautics and Astronautics, Guidance, Navigation, and Control Conference 2002, Reston, Virginia, USA (2002)
15. Schumacher, C.J., Chandler, P.R., Rasmussen, S.J.: Task allocation for wide area search munitions with variable path length. In: Institute of Electrical and Electronic Engineers, American Control Conference 2003, New York, USA (2003)
16. Smith, T.R., Hanssmann, H., Leonard, N.E.: Orientation control of multiple underwater vehicles with symmetry-breaking potentials. In IEEE Conf. Decis. Control. **5**, 4598–4603 (2001)
17. Swaroop, D., Hedrick, J.K.: Constant spacing strategies for platooning in automated highway systems. J. Dyn. Syst. Meas. Control. **121**, 462 (1999)
18. Wolfe, J.D., Chichka, D.F., Speyer, J.L.: Decetntralized controllers for unmanned aerial vehicle formation flight. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, 96–3833, San Diego, California (1996)

19. Yamagishi, M.: Social rules for reactive formation switching. Technical Report UWEETR-2004-0025. Department of Electrical Engineering, University of Washington, Seattle, Washington, USA (2004)
20. Yamaguchi, H.: A cooperative hunting behavior by mobile-robot troops. Int. J. Robotic. Res. **18**(9), 931 (1999)
21. Yamaguchi, H., Arai, T., Beni, G.: A distributed control scheme for multiple robotic vehicles to make group formations. Robotic. Autonom. Syst. **36**(4), 125–147 (2001)

Computational Studies of Randomized Multidimensional Assignment Problems

Mohammad Mirghorbani, Pavlo Krokhmal, and Eduardo L. Pasiliao

Abstract In this chapter, we consider a class of combinatorial optimization problems on hypergraph matchings that represent multidimensional generalizations of the well-known linear assignment problem (LAP). We present two algorithms for solving randomized instances of MAPs with linear and bottleneck objectives that obtain solutions with guaranteed quality.

Keywords Multidimensional assignment problem • Hypergraph matching problem • Probabilistic analysis

1 Introduction

In the simplest form of the assignment problem, two sets V and W with size $|V| = |W| = n$ are given. The goal is to find a permutation of the elements of W, $\pi = (j_1, j_2, \ldots, j_n)$, where the ith element of V is assigned to the element $j_i = \pi(i)$ from W in such a way that the cost function $\sum_{i=1}^{n} a_{i\pi(i)}$ is minimized. Here, a_{ij} is the cost of assigning element i of V to the element j of W. This problem is widely known as the classical linear assignment problem (LAP). The LAP can be

M. Mirghorbani
Department of Mechanical and Industrial Engineering, The University of Iowa,
801 Newton Road Iowa City, IA 52246, USA
e-mail: smirghor@engineering.uiowa.edu

P. Krokhmal (✉)
Department of Mechanical and Industrial Engineering, The University of Iowa,
3131 Seamans Center, Iowa City, IA 52242, USA
e-mail: krokhmal@engineering.uiowa.edu

E.L. Pasiliao
Air Force Research Lab, Eglin AFB, 101 West Eglin. Boulevard, Eglin AFB, FL, USA
e-mail: eduardo.pasiliao@eglin.af.mil

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_12,

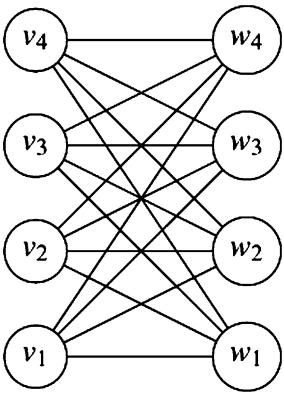

Fig. 1 The underlying bi-partite graph for an assignment problem with $n = 4$

represented by a complete weighted bipartite graph $G = (V, W; E)$, with node sets V and W, where $|V| = |W| = n$ and weight a_{ij} for the edge $(v_i, w_j) \in E$ (Fig. 1), such that an optimal solution for LAP corresponds to a minimum-weight matching in the bipartite graph G. The LAP is well known to be polynomially solvable in $O(n^3)$ time using the celebrated Hungarian method [11]. A mathematical programming formulation of the LAP reads as

$$\begin{aligned} L_n^* = \min_{x_{ij} \in \{0,1\}} & \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij} \\ \text{s. t.} \quad & \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \ldots, n, \\ & \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, n, \end{aligned} \tag{1}$$

where it is well known that the integrality of variables x_{ij} can be relaxed: $0 \le x_{ij} \le 1$. The LAP also admits the following permutation-based formulation:

$$\min_{\pi \in \Pi} \sum_{i=1}^{n} a_{i\pi(i)}, \tag{2}$$

where Π is the set of all permutations of the set $\{1, \ldots, n\}$.

Multidimensional extensions of the bipartite graph matching problems, such as the LAP, quadratic assignment problem (QAP), and so on, can be presented in the framework of *hypergraph matching problems*.

A *hypergraph* $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, also called a *set system*, is a generalization of the graph concept, where a *hyperedge* may connect two or more vertices from the set $\mathcal{V}$:

$$\mathcal{E} = \{e \subset \mathcal{V} \mid |e| \ge 2\}, \tag{3}$$

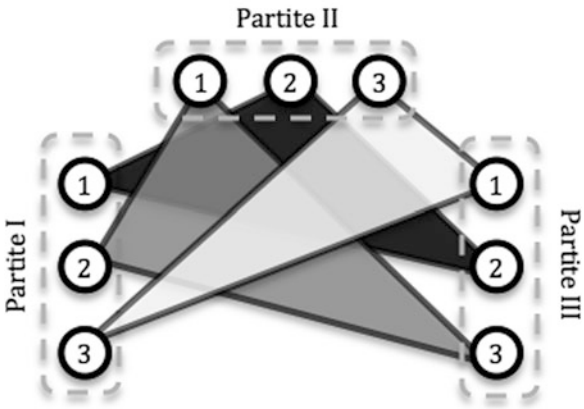

Fig. 2 A perfect matching in a 3-partite 3-uniform hypergraph

A hypergraph is called *k-uniform* if all its hyperedges have the size k:

$$\mathcal{E} = \{e \in \mathcal{V} \mid |e| = k\}.$$

Observe that a regular graph is a 2-uniform hypergraph. A subset $\mathcal{V}' \subset \mathcal{V}$ of vertices is called *independent* if the vertices in $\mathcal{V}'$ do not share any edges; if $\mathcal{V}$ can be partitioned into d independent subsets, $\mathcal{V} = \cup_{k=1}^{d} \mathcal{V}_k$, then $\mathcal{V}$ is called d-partite.

Let $\mathcal{H}_{d|n}$ be a complete d-partite n-uniform hypergraph, where each independent set $\mathcal{V}_k$ has n vertices. Then $\left|\mathcal{V}(\mathcal{H}_{d|n})\right| = n \times d$, and the total number of hyperedges is equal to n^d. A *perfect matching* μ on $\mathcal{H}_{d|n}$ is formed by a set of n hyperedges that do not share any vertices:

$$\mu = \left\{\{e_1, \ldots, e_n\} \mid e_i \in \mathcal{E},\ \ e_i \cap e_j = \emptyset,\ \ i, j \in \{1, \ldots, n\}, i \neq j\right\}.$$

Figure 2 shows a perfect matching in a 3-partite 3-uniform hypergraph.

If the cost of hypergraph matching μ is given by function $\Phi(\mu)$, the general combinatorial optimization problem on hypergraph matchings can be stated as

$$\min\left\{\Phi(\mu) \mid \mu \in \mathcal{M}(\mathcal{H}_{d|n})\right\}, \tag{4}$$

where $\mathcal{M}(\mathcal{H}_{d|n})$ is the set of all perfect matchings on $\mathcal{H}_{d|n}$.

The mathematical programming formulation of the hypergraph matching problem (4) is also generally known as *multidimensional assignment problem (MAP)*. To derive the mathematical programming formulation of (4), note that according to the definition of $\mathcal{H}_{d|n}$, each of its hyperedges contains exactly one vertex from each of the independent sets $\mathcal{V}_1, \ldots, \mathcal{V}_d$ and therefore can be represented as a vector $(i_1, \ldots, i_d) \in \{1, \ldots, n\}^d$, where, with abuse of notation, the set $\{1, \ldots, n\}$ is used to label the nodes of each independent subset $\mathcal{V}_k$. Then, the set $\mathcal{M}(\mathcal{H}_{d|n})$ of perfect matchings on $\mathcal{H}_{d|n}$ can be represented in a mathematical programming form as

$$\mathcal{M}(\mathcal{H}_{d|n}) = \left\{ x \in \{0,1\}^{n^d} \;\middle|\; \sum_{\substack{i_k \in \{1,\dots,n\} \\ k \in \{1,\dots,d\} \setminus \{r\}}} x_{i_1 \cdots i_d} = 1, \;\; i_r \in \{1,\dots,n\}, \right.$$

$$\left. r \in \{1,\dots,d\} \right\}, \tag{5}$$

where $x_{i_1 \cdots i_d} = 1$ if the hyperedge $(i_1, \dots, i_d)$ is included in the matching, and $x_{i_1 \cdots i_d} = 0$ otherwise.

Depending on the particular form of Φ, a number of combinatorial optimization problems on hypergraph matchings can be formulated. For instance, if the cost function Φ in (4) is defined as a linear form over the variables $x_{i_1 \cdots i_d}$,

$$\Phi(x) = \sum_{i_1=1}^{n} \cdots \sum_{i_d=1}^{n} \phi_{i_1 \cdots i_d} \, x_{i_1 \cdots i_d}, \tag{6}$$

one obtains the so-called linear multidimensional assignment problem (LMAP):

$$\begin{aligned} Z^*_{d,n} = \min_{x \in \{0,1\}^{n^d}} & \sum_{i_1=1}^{n} \cdots \sum_{i_d=1}^{n} \phi_{i_1 \cdots i_d} x_{i_1 \cdots i_d} \\ \text{s. t.} \quad & \sum_{i_2=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, && i_1 = 1, \dots, n, \\ & \sum_{i_1=1}^{n} \cdots \sum_{i_{k-1}=1}^{n} \sum_{i_{k+1}=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, && i_k = 1, \dots, n, \\ & && k = 2, \dots, d-1, \\ & \sum_{i_1=1}^{n} \cdots \sum_{i_{d-1}=1}^{n} x_{i_1 \cdots i_d} = 1, && i_d = 1, \dots, n. \end{aligned} \tag{7}$$

Clearly, a special case of (7) with $d = 2$ is nothing else but the classical LAP (1). The *dimensionality* parameter d in (7) stands for the number of "dimensions" of the problem, or sets of elements that need to be assigned to each other, while the parameter n is known as the *cardinality* parameter.

If the cost of the matching on hypergraph $\mathcal{H}_{d|n}$ is defined as the cost of the most expensive hyperedge in the matching, i.e., the cost function $\Phi(x)$ has the form

$$\Phi(x) = \max_{i_1,\dots,i_d \in \{1,\dots,n\}} \phi_{i_1 \cdots i_d} \, x_{i_1 \cdots i_d},$$

we obtain the multidimensional assignment problem with bottleneck objective (BMAP):

$$
\begin{aligned}
W^*_{d,n} = \min_{x\in\{0,1\}^{n^d}} \max_{i_1,\ldots,i_d\in\{1,\ldots,n\}} & \phi_{i_1\cdots i_d}\, x_{i_1\cdots i_d} \\
\text{s. t.} \quad & \sum_{i_2=1}^{n}\cdots\sum_{i_d=1}^{n} x_{i_1\cdots i_d} = 1, && i_1 = 1,\ldots,n, \\
& \sum_{i_1=1}^{n}\cdots\sum_{i_{k-1}=1}^{n}\sum_{i_{k+1}=1}^{n}\cdots\sum_{i_d=1}^{n} x_{i_1\cdots i_d} = 1, && i_k = 1,\ldots,n, \\
& && k = 2,\ldots,d-1, \\
& \sum_{i_1=1}^{n}\cdots\sum_{i_{d-1}=1}^{n} x_{i_1\cdots i_d} = 1, && i_d = 1,\ldots,n. \qquad (8)
\end{aligned}
$$

Similarly, taking the hypergraph matching cost function Φ in (4) as a quadratic form over $x \in \{0,1\}^{n^d}$,

$$
\Phi(x) = \sum_{i_1=1}^{n}\cdots\sum_{i_d=1}^{n}\sum_{j_1=1}^{n}\cdots\sum_{j_d=1}^{n} \phi_{i_1\cdots i_d\, j_1\cdots j_d}\, x_{i_1\cdots i_d}\, x_{j_1\cdots j_d}, \qquad (9)
$$

we arrive at the quadratic multidimensional assignment problem (QMAP), which represents a higher-dimensional generalization of the classical QAP.

The LMAP was first introduced by Pierskalla [12], and has found applications in the areas of data association, sensor fusion, multisensor multi-target tracking, peer-to-peer refueling of space satellites, etc. for a detailed discussion of the applications of the LMAP, see, e.g., [3, 4]. In [2], a two-step method based on bipartite and multidimensional matching problem is proposed to solve the roots of a system of polynomial equations that avoid possible degeneracies and multiple roots encountered in some conventional methods. MAP is used in the course timetabling problem, where the goal is to assign students and teachers to classes and time slots [5]. In [1] a composite neighborhood structure with a randomized iterative improvement algorithm for the timetabling problem with a set of hard and soft constraints is proposed. An application of MAP in the scheduling of sport competitions that take place in different venues is studied in [15]. The characteristic of this study is that venues, that can involve playing fields, courts, or drill stations, are considered as part of the scheduling process. In [13] a Lagrangian relaxation based algorithm is proposed for the multi-target/multisensor tracking problem, where multiple sensors are used to identify targets and estimate their states. To accurately achieve this goal, the data association problem which is an NP-hard problem should be solved to partition observations into tracks and false alarms. A general class of these data association problems can be formulated as a multidimensional assignment problem with a Bayesian estimation as the objective function. The optimal solution yields the maximum a posteriori estimate. A special case of multiple-target tracking problem is studied in [14] to track the flight paths of charged elementary particles near to their primary point of interaction. The three-dimensional assignment problem is used in [6] to formulate a peer-to-peer

(P2P) satellite refueling problem. P2P strategy is an alternative to the single vehicle refueling system where all satellites share the responsibility of refueling each other on an equal footing.

The remainder of this chapter is organized as follows: In Sect. 2, heuristic methods to solve multidimensional assignment problems will be provided. Section 2.1 describes the method to solve MAPs with large cardinality. In Sect. 2.2, the heuristic method for MAPs with large dimensionality is explained. Section 3 contains the numerical results and comparison with exact methods, and finally in Sect. 4, conclusions and future extensions are provided.

2 High-quality Solution Sets in Randomized Multidimensional Assignment Problems

In this section two methods will be described that can be used to obtain mathematically proven high-quality solutions for MAPs with large cardinality or large dimensionality. These methods utilize the concept of *index graph* of the underlying hypergraph of the problem.

2.1 Random Linear MAPs of Large Cardinality

In the case when the cost Φ of hypergraph matching is a linear function of hyperedges' costs, i.e., for MAPs with linear objectives, a useful tool for constructing high-quality solutions for instances with large cardinality ($n \gg 1$) is the so-called *index graph*. The index graph is related to the concept of *line graph* in that the vertices of the index graph represent the hyperedges of the hypergraph.

Namely, by indexing each vertex of the index graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ by $(i_1, \ldots, i_d) \in \{1, \ldots, n\}^d$, identically to the corresponding hyperedge of $\mathcal{H}_{d|n}$, the set of vertices $\mathcal{V}^*$ can be partitioned into n subsets $\mathcal{V}_k^*$, also called *levels*, which contain vertices whose first index is equal to k:

$$\mathcal{V}^* = \bigcup_{k=1}^{n} \mathcal{V}_k^*, \quad \mathcal{V}_k^* = \{(k, i_2, \ldots, i_d) \mid i_2, \ldots, i_d \in \{1, \ldots, n\}\}.$$

For any two vertices $i, j \in \mathcal{V}^*$, an edge (i, j) exists in $\mathcal{G}^*$, $(i, j) \in \mathcal{E}^*$, if and only if the corresponding hyperedges of $\mathcal{H}_{d|n}$ do not have common nodes (Fig. 3). In other words,

$$\mathcal{E}^* = \{(i, j) \mid i = (i_1, \ldots, i_d), j = (j_1, \ldots, i_d) : i_k \neq j_k,\ k = 1, \ldots, n\}.$$

Then, that it is easy to see that $\mathcal{G}^*$ has the following properties.

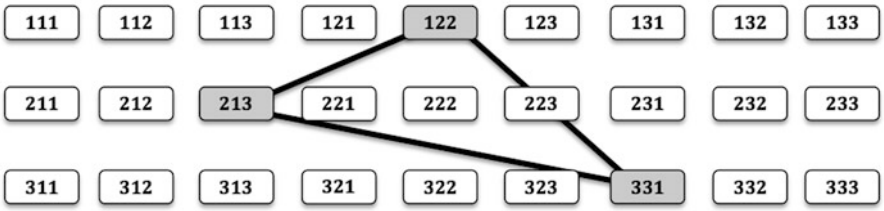

Fig. 3 The index graph $\mathcal{G}^*$ of the hypergraph $\mathcal{H}_{d|n}$ shown in Fig. 2. The vertices of $\mathcal{G}^*$ shaded in *gray* represent a clique (or, equivalently, a perfect matching on $\mathcal{H}_{d|n}$)

Lemma 1. *Consider a complete, d-partite, n-uniform hypergraph $\mathcal{H}_{d|n} = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{E}| = n^d$, and $\mathcal{V} = \bigcup_{k=1}^{d} \mathcal{V}_k$ such that $\mathcal{V}_k \cap \mathcal{V}_l = \emptyset$, $k \neq l$, and $|\mathcal{V}_k| = n$, $k = 1, \ldots, d$. Then, the index graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ of $\mathcal{H}_{d|n}$ satisfies:*

1. *$\mathcal{G}^*$ is n-partite, namely $\mathcal{V}^* = \bigcup_{k=1}^{n} \mathcal{V}_k^*$, $\mathcal{V}_i^* \cap \mathcal{V}_j^* = \emptyset$ for $i \neq j$, where each $\mathcal{V}_k^*$ is an independent set in $\mathcal{V}^*$: for any $i, j \in \mathcal{V}_k^*$ one has $(i, j) \notin \mathcal{E}^*$.*
2. *$|\mathcal{V}_k^*| = n^{d-1}$ for each $k = 1, \ldots, n$.*
3. *The set of perfect matchings in $\mathcal{H}_{d|n}$ is isomorphic to the set of n-cliques in $\mathcal{G}^*$, i.e., each perfect matching in $\mathcal{H}_{d|n}$ corresponds uniquely to a (maximum) clique of size n in $\mathcal{G}^*$.*

Let us denote by $\mathcal{G}^*(\alpha_n)$ the induced subgraph of the index graph $\mathcal{G}^*$ obtained by *randomly* selecting α_n vertices from each level $\mathcal{V}_k^*$ of $\mathcal{G}^*$, and also define $N(\alpha_n)$ to be the number of cliques in $\mathcal{G}^*(\alpha_n)$, then based on the following lemma [9] one can select α_n in such a way that $\mathcal{G}^*(\alpha_n)$ is expected to contain at least one n-clique:

Lemma 2. *The subgraph $\mathcal{G}^*(\alpha_n)$ is expected to contain at least one n-clique, or a perfect matching on $\mathcal{H}_{d|n}$ (i.e., $E[N(\alpha_n)] \geq 1$) when α_n is equal to*

$$\alpha_n = \left\lceil \frac{n^{d-1}}{n!^{\frac{d-1}{n}}} \right\rceil. \tag{10}$$

In the case when the cost coefficients $\phi_{i_1 \cdots i_d}$ of MAP with linear or bottleneck objective are drawn independently from a given probability distribution, Lemma 2 can be used to construct high-quality solutions. The approach is to create the subgraph $\mathcal{G}^*_{\min}(\alpha_n)$, also called the *$\alpha$-set*, from the index graph $\mathcal{G}^*$ of the MAP by selecting α_n nodes with the smallest cost coefficients from each partition (level) of $\mathcal{G}^*$. If the costs of the hyperedges of $\mathcal{H}_{d|n}$, or, equivalently, vertices of $\mathcal{G}^*$, are identically and independently distributed, the α-set is expected to contain at least one clique, which represents a perfect matching in the hypergraph $\mathcal{H}_{d|n}$ (Fig. 2). It should be noted that since the α-set is created from the nodes with the smallest cost coefficients, if a clique exists in the α-set, the resulting cost of the perfect matching is expected to be close to the optimal solution of the MAP.

Importantly, when the cardinality n of the MAP increases, the size of the subgraph $\mathcal{G}^*(\alpha_n)$ or $\mathcal{G}^*_{\min}(\alpha_n)$ grows only as $O(n)$, as evidenced by the following observation:

Lemma 3. *If d is fixed and $n \to \infty$, then α_n monotonically approaches a finite limit:*

$$\alpha_n \nearrow \alpha := \lceil e^{d-1} \rceil \quad as \quad n \nearrow \infty. \tag{11}$$

Corollary 1. *In the case of randomized MAP of large enough cardinality $n \gg 1$ the subset $\mathcal{G}^*_{\min}$ expected to contain a high-quality feasible solution of the MAP can simply be chosen as $\mathcal{G}^*_{\min}(\alpha)$, where α is given by* (11)*.*

Observe that using the α-set $\mathcal{G}^*_{\min}(\alpha)$ for construction of a low-cost feasible solution to randomized MAP with linear or bottleneck objectives may prove to be a challenging task, since it is equivalent to finding an n-clique in an n-partite graph; moreover, the graph $\mathcal{G}^*_{\min}(\alpha)$ is only expected to contain a single n-clique (feasible solution). The following variation of Lemma 2 allows for constructing a subgraph of $\mathcal{G}^*$ that contains exponentially many feasible solutions:

Lemma 4. *Consider the index graph $\mathcal{G}^*$ of the underlying hypergraph $\mathcal{H}_{d|n}$ of a randomized MAP, and let*

$$\beta_n = \left\lceil 2 \frac{n^{d-1}}{n!^{\frac{d-1}{n}}} \right\rceil. \tag{12}$$

Then, the subgraph $\mathcal{G}^*(\beta_n)$ is expected to contain 2^n n-cliques, or, equivalently, perfect matching on $\mathcal{H}_{d|n}$.

Proof. The statement of the lemma is easy to obtain by regarding the feasible solutions of the MAP as *paths* that contain exactly one vertex in each of the n "levels" $\mathcal{V}^*_1, \ldots, \mathcal{V}^*_n$ of the index graph $\mathcal{G}^*$. Namely, let us call a path connecting the vertices $\left(1, i_2^{(1)}, \ldots, i_d^{(1)}\right) \in \mathcal{V}^*_1$, $\left(2, i_2^{(2)}, \ldots, i_d^{(2)}\right) \in \mathcal{V}^*_2, \ldots, \left(n, i_2^{(n)}, \ldots, i_d^{(n)}\right) \in \mathcal{V}^*_n$ *feasible* if $\left\{i_k^{(1)}, i_k^{(2)}, \ldots, i_k^{(n)}\right\}$ is a permutation of the set $\{1, \ldots, n\}$ for every $k = 2, \ldots, d$. Note that from the definition of the index graph $\mathcal{G}^*$ it follows that a path is feasible if and only if the vertices it connects form an n-clique in $\mathcal{G}^*$. Next, observe that a path in $\mathcal{G}^*$ chosen at random is feasible with the probability $\left(\frac{n!}{n^n}\right)^{d-1}$, since one can construct $n^{n(d-1)}$ different (not necessarily feasible) paths in $\mathcal{G}^*$. Then, if we randomly select β_n vertices from each set $\mathcal{V}^*_k$ in such a way that out of the $(\beta_n)^n$ paths spanned by $\mathcal{G}^*(\beta_n)$ at least 2^n are feasible, the value of β_n must satisfy:

$$(\beta_n)^n \left(\frac{n!}{n^n}\right)^{d-1} \geq 2^n,$$

from which it follows immediately that β_n must satisfy (12).

Corollary 2. *If d is fixed and $n \to \infty$, then β_n monotonically approaches a finite limit:*

$$\beta_n \nearrow \beta := \lceil 2e^{d-1} \rceil \quad as \quad n \nearrow \infty.$$

Remark 1. Since the value of the parameter β_n (12) is close to the double of the parameter α_n (10), the subgraph $\mathcal{G}^*_{\min}(\beta_n)$, constructed from selecting β_n nodes with the smallest cost coefficients from each partition (level) of $\mathcal{G}^*$ will be called the "2α-set," or $\mathcal{G}^*(2\alpha)$.

Following [10], the costs of feasible solutions of randomized MAPs with linear or bottleneck objectives that are contained in the α- or 2α-sets can be shown to satisfy:

Lemma 5. *Consider a randomized MAP with linear or bottleneck objectives, whose cost coefficients are iid random variables from a continuous distribution F with a finite left endpoint of the support, $F^{-1}(0) > -\infty$. Then, for a fixed $d \geq 3$ and large enough values of n, if the subset $\mathcal{G}^*_{\min}(\alpha)$ (or, respectively, $\mathcal{G}^*_{\min}(\beta)$) contains a feasible solution of the MAP, the cost Z_n of this solution satisfies*

$$(n-1)F^{-1}(0) + F^{-1}\left(\frac{1}{n^{d-1}}\right) \leq Z_n \leq nF^{-1}\left(\frac{3\ln n}{n^{d-1}}\right), \quad n \gg 1, \tag{13}$$

in the case of MAP with linear objective (7), while in the case of MAP with bottleneck objective (8) the cost W_n of such a solution satisfies

$$F^{-1}\left(\frac{1}{n^{d-1}}\right) \leq W_n \leq F^{-1}\left(\frac{3\ln n}{n^{d-1}}\right), \quad n \gg 1. \tag{14}$$

2.2 Random MAPs of Large Dimensionality

In cases where the cardinality of the MAP is fixed, and its dimensionality is large, $d \gg 1$, the approach described in Sect. 2.1 based on the construction of α- or 2α-subset of the index graph $\mathcal{G}^*$ of the MAP is not well suited, since in this case the size of $\mathcal{G}^*(\alpha)$ grows exponentially in d.

However, the index graph $\mathcal{G}^*$ of the underlying hypergraph $\mathcal{H}_{d|n}$ of the MAP can still be utilized to construct high-quality solutions of large-dimensionality randomized MAPs.

Let us call two matchings $\mu_i = \left\{\left(i_1^{(1)}, \ldots, i_d^{(1)}\right), \ldots, \left(i_1^{(n)}, \ldots, i_d^{(n)}\right)\right\}$ and $\mu_j = \left\{\left(j_1^{(1)}, \ldots, j_d^{(1)}\right), \ldots, \left(j_1^{(n)}, \ldots, j_d^{(n)}\right)\right\}$ on the hypergraph $\mathcal{H}_{d|n}$ *disjoint* if

$$\left(i_1^{(k)}, \ldots, i_d^{(k)}\right) \neq \left(j_1^{(\ell)}, \ldots, j_d^{(\ell)}\right) \quad \text{for all} \quad 1 \leq k, \ell \leq n,$$

or, in other words, if μ_i and μ_j do not have any common hyperedges. It is easy to see that if the cost coefficients of randomized MAPs are iid random variables, then the costs of the feasible solutions corresponding to the disjoint matchings are also independent and identically distributed.

Next, we show how the index graph $\mathcal{G}^*$ of the MAP can be used to construct exactly n^{d-1} disjoint solutions whose costs are iid random variables. First, recalling the interpretation of feasible MAP solutions as *paths* in the index graph $\mathcal{G}^*$, we observe that disjoint solutions of MAP, or, equivalently, disjoint matchings on $\mathcal{H}_{d|n}$ are represented by disjoint paths in $\mathcal{G}^*$ that do not have common vertices.

Note that since each level $\mathcal{V}_k^*$ of $\mathcal{G}^*$ contains exactly n^{d-1} vertices (see Lemma 1), there may be no set of disjoint paths with more than n^{d-1} elements.

On the other hand, recall that a (feasible) path $\mathcal{G}^*$ can be described as a set of n vectors

$$\mu = \left\{ \left(i_1^{(1)}, \ldots, i_d^{(1)}\right), \ldots, \left(i_1^{(n)}, \ldots, i_d^{(n)}\right) \right\},$$

such that $\left\{i_k^{(1)}, \ldots, i_k^{(n)}\right\}$ is a permutation of the set $\{1, \ldots, n\}$ for each $k = 1, \ldots, d$. Then, for any given vertex $v^{(1)} = \left(1, i_2^{(1)}, \ldots, i_d^{(1)}\right) \in \mathcal{V}_1^*$, let us construct a feasible path containing $v^{(1)}$ in the form

$$\left\{ \left(1, i_2^{(1)}, \ldots, i_d^{(1)}\right), \left(2, i_2^{(2)}, \ldots, i_d^{(2)}\right), \ldots, \left(n, i_2^{(n)} \cdots i_d^{(n)}\right) \right\},$$

where for $k = 2, \ldots, d$ and $r = 2, \ldots, n$

$$i_k^{(r)} = \begin{cases} i_k^{(r-1)} + 1, \text{ if } i_k^{(r-1)} = 1, \ldots, n-1, \\ \qquad 1, \text{ if } i_k^{(r-1)} = n. \end{cases} \tag{15}$$

In other words, $\left\{i_k^{(1)}, \ldots, i_k^{(n)}\right\}$ is a forward cyclic permutation of the set $\{1, \ldots, n\}$ for any $k = 2, \ldots, d$. Applying (15) to each of the n^{d-1} vertices $\left(1, i_2^{(1)}, \ldots, i_d^{(1)}\right) \in \mathcal{V}_1^*$, we obtain n^{d-1} feasible paths (matchings on $\mathcal{H}_{d|n}$) that are mutually disjoint, since (15) defines a bijective mapping between any vertex (hyperedge) $\left(k, i_2^{(k)}, \ldots, i_d^{(k)}\right)$ from the set $\mathcal{V}_k^*, k = 2, \ldots, n$, and the corresponding vertex (hyperedge) $v^{(1)} \in \mathcal{V}_1^*$.

Then, if hyperedge costs $\phi_{i_1 \ldots i_d}$ in the linear or bottleneck MAPs (7) and (8) are stochastically independent, the costs $\Phi(\mu_1), \ldots, \Phi(\mu_{n^{d-1}})$ of the n^{d-1} disjoint matchings $\mu_1, \ldots, \mu_{n^{d-1}}$ defined by (15) are also independent, as they do not contain any common elements $\phi_{i_1 \ldots i_d}$. Given that the optimal solution cost $Z_{d,n}^*$ (respectively, $W_{d,n}^*$) of randomized linear (respectively, bottleneck) MAP does not exceed the costs $\Phi(\mu_1), \ldots, \Phi(\mu_{n^{d-1}})$ of the disjoint solutions described by (15), the following bound on the optimal cost of linear or bottleneck randomized MAP can be established.

Lemma 6. *The optimal costs $Z_{d,n}^*$, $W_{d,n}^*$ of random MAPs with linear or bottleneck objectives (7), (8), where cost coefficients are iid random variables, satisfy*

$$Z_{d,n}^* \le X_{1:n^{d-1}}^{\sum}, \quad W_{d,n}^* \le X_{1:n^{d-1}}^{\max}, \tag{16}$$

where $X_i^{\sum}$, $X_i^{\max}$ $(i = 1, \ldots, n^{d-1})$ *are iid random variables with distributions* $F^{\sum,\max}$ *that are determined by the form of the corresponding objective function, and* $X_{1:k}$ *denotes the minimum-order statistic among* k *iid random variables.*

Remark 2. Inequalities in (16) are tight: namely, in the special case of random MAPs with $n = 2$, all of the $n!^{d-1} = 2^{d-1}$ feasible solutions are stochastically independent [7], whereby equalities hold in (16).

As shown in [10], the following quality guarantee on the minimum cost of the n^{d-1} disjoint solutions (15) of linear and bottleneck MAPs can be established:

$$X^{\sum}_{1:n^{d-1}} \leq nF^{-1}\left(n^{-\frac{d-1}{2n}}\right), \quad X^{\max}_{1:n^{d-1}} \leq F^{-1}\left(n^{-\frac{d-1}{2n}}\right), \quad d \gg 1,$$

where F^{-1} is the inverse of the distribution function F of the cost coefficients $\phi_{i_1 \cdots i_d}$. This observation allows for constructing high-quality solutions of randomized linear and bottleneck MAPs by searching the set of disjoint feasible solutions as defined by (15).

3 Numerical Results

Sections 2.1 and 2.2 introduced two methods of solving randomized instances of MAPs by constructing subsets (neighborhoods) of the feasible set of the problem that are guaranteed to contain high-quality solutions whose costs approach optimality when the problem size ($n \to \infty$, or, respectively, $d \to \infty$) increases. In this section we investigate the quality of solutions contained in these neighborhoods for small- to moderate-sized problem instances and compare the results with the optimal solutions where it is possible.

Before proceeding with the numerical results of the study, in the next section, FINDCLIQUE, the algorithm that is used to find the optimum clique in the index-graph $\mathcal{G}^*$ or the first clique in the α-set or 2α-set will be described. The results from randomly generated MAP instances for each of these two methods are presented next.

3.1 Finding n-Cliques in n-Partite Graphs

In order to find cliques in $\mathcal{G}^*$, the α-set, or the 2α-set, the branch-and-bound algorithm proposed in [8] is used. This algorithm, called FINDCLIQUE, is designed to find all n-cliques contained in an unweighed n-partite graph.

The input to original FINDCLIQUE is an n-partite graph $G(V_1, \ldots, V_n; E)$ with the adjacency matrix $\mathsf{M} = (m_{ij})$, and the output will be a list of all n-cliques

contained in G. Nodes from G are copied into a set called `compatible nodes`, denoted by C. The set C is further divided into n partitions, each denoted by C_i that are initialized such that they contain nodes from partite V_i, $i = \{1, \ldots, n\}$. FINDCLIQUE also maintains two other sets, namely, `current clique`, denoted by Q and `erased nodes`, denoted by E. The set Q holds a set of nodes that are pairwise adjacent and construct a clique. The `erased node` set, E, is furthered partitioned into n sets, denoted by E_i, that are initialized as empty. At each step of the algorithm, E_i will contain the nodes that are not adjacent to the ith node added to Q.

The branch-and-bound tree has n levels, and FINDCLIQUE searches for n-cliques in the tree in a depth-first fashion. At level t of the branch of bound algorithm, the index of the smallest partition in C, $\theta = \arg\min_i \{|C_i| \,|\, i \notin \mathsf{V}\}$ will be detected, and C_θ will be marked as visited by including θ into $\mathsf{V} \leftarrow \{\mathsf{V} \cup \theta\}$, where V is the list of partitions that have a node in Q. Then, a node q from C_θ is selected at random and added to Q. If $|Q| = n$, an n-clique is found. Otherwise, C will be updated; every partition C_i where $i \notin \mathsf{V}$ will be searched for nodes c_{ij}, $(j = 1, \ldots, |C_i|)$ that are not adjacent to q, i.e., $m_{q,c_{ij}} = 0$. Any such node will be removed from C_i and will be transferred to E_t. Note that in contrast to C, nodes in different levels of E will not necessarily be from the same partite of G. Decision regarding backtracking is made after C is updated. It is obvious that in an n-partite graph the following will hold:

$$\omega(G) \leq n, \tag{17}$$

where $\omega(G)$ is the size of a maximum clique in G. In other words, the size of any maximum clique cannot be larger than the number of partites in that the maximum clique can only contain at most one node from each partite of G. If after updating, there is any $C_i \notin \mathsf{V}$ with $|C_i| = 0$, adding q_i to Q will not result in a clique of size n, since the condition in (17) changes into strict inequality. In such cases, q is removed from Q, nodes from E_t will be transferred back to their respective partitions in C, and FINDCLIQUE will try to add another node from C_θ that is not already branched on, to Q. If such a node does not exist, the list of visited partitions will be updated ($\mathsf{V} \leftarrow \mathsf{V} \backslash \theta$), and FINDCLIQUE backtracks to the previous level of the branch-and-bound tree. If the backtracking condition is not met and q is promising, FINDCLIQUE will go one level deeper in the tree, finds the next smallest partition in the updated C and tries to add a new node to Q.

When solving the clique problem in the α-set or 2α-set, since the objective is to find the first n-clique regardless of its cost, FINDCLIQUE can be used without any modifications, and the weights of the nodes in $\mathcal{G}^*_{\min}(\alpha)$ or $\mathcal{G}^*_{\min}(2\alpha)$ will be ignored. However, when the optimal clique with the smallest cost in $\mathcal{G}^*$ is sought, some modifications in FINDCLIQUE are necessary to enable it to deal with weighted graphs. The simplest way to adjust FINDCLIQUE is to compute the weight of the n-cliques as they are found, and report the clique with the smallest cost as the output of the algorithm. This is the method that is used in the experimental studies whenever the optimal solution is desired. However, to obtain a more efficient

algorithm, it is possible to calculate the weight of the partial clique contained in Q in every step of the algorithm and fathom subproblems for which $W_Q \geq W_{Q^*}$, where W_Q and W_{Q^*} are the cost of the partial clique in Q and the cost of the best clique found so far by the algorithm, respectively. Further improvement can be achieved by sorting the nodes in C_i, $i = 1, \ldots, n$, based on their cost coefficients, and each time select the untraversed node with the smallest node as the next node to be added to Q (as opposed to randomly selecting a node, which does not change the overall computational time in the unweighted graph if a list of all n-cliques is desired). This enables us to compute a lower bound on the cost of the maximum clique that the nodes in Q may lead to as follows:

$$\mathrm{LB}_Q = W_Q + \sum_{i \notin \mathsf{V}} w_i^{\min}, \tag{18}$$

where $w_i^{\min}$ is the weight of the node with the smallest cost coefficient in C_i. Any subproblem with $\mathrm{LB}_Q \geq W_{Q^*}$ will be fathomed.

3.2 Random Linear MAPs of Large Cardinality

To demonstrate the performance of the method described in Sect. 2.1, random MAPs with fixed dimensionality $d = 3$ and different values of cardinality n are generated. The cost coefficients $\phi_{i_1 \cdots i_d}$ are randomly drawn from the uniform $U[0, 1]$ distribution. Three sets of problems are solved for this case: (i) $n = 3, \ldots, 8$ with $d = 3$, solved for optimality, and the first clique in the α- and 2α-sets, (ii) $n = 10, 15, \ldots, 45$, with $d = 3$, solved for the first clique in the α- and 2α-sets, and finally (iii) $n = 50, 55, \ldots, 80$, with $d = 3$, solved for the first clique in the 2α-set. For each value of n, 25 instances are generated and solved by modified FINDCLIQUE for the optimum clique or FINDCLIQUE whenever the first clique in the problem is desired. Algorithm is terminated if the computational time needed to solve an instance exceeds 1 h.

In the first group, (i), instances of MAP that admit solution to optimality in a reasonable time were solved. The results from this subset are used to determine the applicability of Corollary 1 and bounds (13) and (14) for relatively small values of n. Table 1 summarizes the average values for the cost of the clique and computational time needed for MAPs with the linear sum objective function for the instances in group (i). The first column, n, is the cardinality of the problem. The columns under the heading "Exact" contain the values related to the optimal clique in $\mathcal{G}^*$. The columns under the heading "$\mathcal{G}^*_{\min}(\alpha_n)$" represent the values obtained from solving the α-set for the first clique, and those under the heading "$\mathcal{G}^*_{\min}(2\alpha)$" represent the values obtained from solving the 2α-set for the first clique. For each of these multicolumns, T denotes the average computational time in seconds, Z is the average cost of the cliques, $|V|$ is the order of the graph or induced subgraph in

Table 1 Comparison of the computational time and cost for the optimum clique and the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear sum objective functions for instances in group (i)

	Exact				$\mathcal{G}^*_{\min}(\alpha)$				$\mathcal{G}^*_{\min}(2\alpha)$			
n	$T^*_{n,3}$	$Z^*_{n,3}$	$\lvert V\rvert$	∃ CLQ	$T_{\mathcal{G}_{\min}(\alpha)}$	$Z_{\mathcal{G}_{\min}(\alpha)}$	$\lvert V\rvert$	∃ CLQ	$T_{\mathcal{G}_{\min}(2\alpha)}$	$Z_{\mathcal{G}_{\min}(2\alpha)}$	$\lvert V\rvert$	∃ CLQ
3	0.02	0.604	3×26	100	0.04	0.609	3×3	76	0.03	0.773	3×6	100
4	0.01	0.458	4×63	100	0.03	0.514	4×4	88	0.03	0.635	4×7	100
5	0.02	0.371	5×124	100	0.04	0.399	5×4	72	0.03	0.571	5×8	100
6	0.31	0.374	6×215	100	0.04	0.452	6×5	92	0.01	0.524	6×9	100
7	14.83	0.329	7×342	100	0.04	0.392	7×5	80	0.05	0.47	7×9	100
8	937.67	0.274	8×511	100	0.05	0.329	8×5	72	0.04	0.478	8×10	100

Table 2 Comparison of the computational time and cost for the optimum clique and the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear bottleneck objective functions for instances in group (i)

	Exact				$\mathcal{G}^*_{\min}(\alpha)$				$\mathcal{G}^*_{\min}(2\alpha)$			
n	$T^*_{n,3}$	$W^*_{n,3}$	$\lvert V\rvert$	∃ CLQ	$T_{\mathcal{G}_{\min}(\alpha)}$	$W_{\mathcal{G}_{\min}(\alpha)}$	$\lvert V\rvert$	∃ CLQ	$T_{\mathcal{G}_{\min}(2\alpha)}$	$W_{\mathcal{G}_{\min}(2\alpha)}$	$\lvert V\rvert$	∃ CLQ
3	0.01	0.321	3×26	100	0.03	0.324	3×3	76	0.04	0.439	3×6	100
4	0.01	0.205	4×63	100	0.03	0.241	4×4	88	0.03	0.311	4×7	100
5	0.01	0.151	5×124	100	0.02	0.17	5×4	72	0.03	0.27	5×8	100
6	0.3	0.124	6×215	100	0.04	0.166	6×5	92	0.04	0.219	6×9	100
7	14.96	0.098	7×342	100	0.04	0.131	7×5	80	0.04	0.163	7×9	100
8	956.6	0.075	8×511	100	0.04	0.092	8×5	72	0.04	0.157	8×10	100

$\mathcal{G}^*$, $\mathcal{G}^*_{\min}(\alpha)$, or $\mathcal{G}^*_{\min}(2\alpha)$, and ∃ CLQ shows the percentage of the problems for which the α-set or 2α-set, respectively, contains a clique. This value is 100% for the exact method. There was no instances in group (i) for which the computational time exceeded 1 h.

It is clear that using α-set or 2α-set enables us to obtain a high-quality solution in a much shorter time by merely searching a significantly smaller part of the index graph $\mathcal{G}^*$. Based on the values for Z, the cost of the clique found in α-set or 2α-set are consistently converging to that of the optimal clique and they provide tight upper bounds for the optimum cost. Additionally, as is shown in the $|V|$ column, significant reduction in the size of the graph can be obtained if α-set or 2α-set are used.

Table 2 contains the corresponding results for the case of a random MAP with bottleneck objective. In this table, W represents the value for the cost of the optimal clique or the first clique found in α- or 2α-set. Figure 4(a) shows how the cost of an optimum clique compares to the cost of the clique found in α-set and 2α-set. Clearly, the cost of optimal clique approaches 0 for both linear sum and linear bottleneck MAPs. Figure 4(b) demonstrates the computational time for instances in group (i).

The advantage of using α-set over 2α-set is that the quality of the detected clique is expected to be higher. On average, however, a clique in 2α-set is found in a shorter time than in α-set.

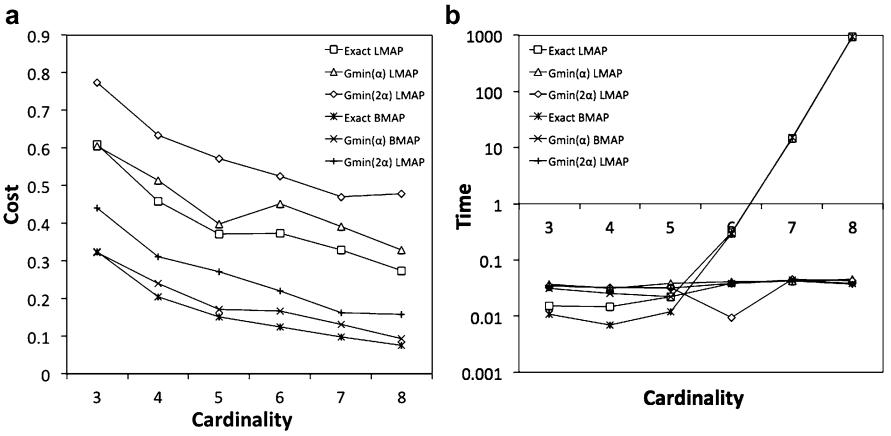

Fig. 4 Solution costs **(a)** and computational time **(b)** in random MAPs with linear sum and linear bottleneck objective functions for instances in group (i)

Table 3 Comparison of the computational time and cost for the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear sum objective functions for instances in group (ii)

	$\mathcal{G}^*_{\min}(\alpha)$					$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{\min}(\alpha)}$	$Z_{\mathcal{G}_{\min}(\alpha)}$	$\lvert V\rvert$	∃ CLQ	Timeout	$T_{\mathcal{G}_{\min}(2\alpha)}$	$Z_{\mathcal{G}_{\min}(2\alpha)}$	$\lvert V\rvert$	∃ CLQ	Timeout
10	0.05	0.266	10×5	60	-	0.05	0.37	10×10	100	-
15	0.06	0.228	15×6	76	-	0.06	0.313	15×11	100	-
20	0.08	0.165	20×6	56	-	0.07	0.246	20×12	100	-
25	0.15	0.147	25×7	80	-	0.08	0.2	25×13	100	-
30	0.89	0.134	30×7	92	-	0.09	0.171	30×13	100	-
35	8.54	0.11	35×7	88	-	0.14	0.151	35×13	100	-
40	100.85	0.097	40×7	92	-	0.46	0.131	40×13	100	-
45	405.16	0.085	45×7	80	16	1.09	0.122	45×14	100	-

The second group of problems, (ii), comprises instances that cannot be solved to optimality within 1 h. The range of n for this group is such that the first clique in the α-set is expected to be found within 1 h. Tables 3 and 4 summarize the results obtained for this group. Instances with $n = 45$ were the largest problems in this group for which α-set could be solved within 1 h. As it is expected, the 2α-set can be solved quickly in a matter of seconds where the equivalent problem for α-set requires a significantly longer computational time. However, the quality of the solutions found for α-set is higher than the quality for solutions in 2α-set. Nonetheless, using 2α-set increases the odds of finding a clique, as based on Lemma 4, 2α-set is expected to contain an exponential number of cliques. It is obvious from the ∃ CLQ column that not all of the instances in α-set contain at least a clique, whereas 100% of the instances in 2α-set contain one that can be found within 1 h. Column *Timeout* represents the percentage of the problems that could not be solved within the allocated 1 h time limit. Out of 25 instances solved for

Table 4 Comparison of the computational time and cost for the first clique found in $\mathcal{G}^*(\alpha)$ and $\mathcal{G}^*(2\alpha)$ in random MAPs with linear bottleneck objective functions for instances in group (ii)

	$\mathcal{G}^*_{\min}(\alpha)$					$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{\min}(\alpha)}$	$W_{\mathcal{G}_{\min}(\alpha)}$	$\lvert V\rvert$	$\exists$ CLQ	Timeout	$T_{\mathcal{G}_{\min}(2\alpha)}$	$W_{\mathcal{G}_{\min}(2\alpha)}$	$\lvert V\rvert$	$\exists$ CLQ	Timeout
10	0.04	0.065	10×5	60	-	0.02	0.098	10×10	100	-
15	0.04	0.037	15×6	76	-	0.02	0.056	15×11	100	-
20	0.05	0.023	20×6	56	-	0.04	0.036	20×12	100	-
25	0.1	0.017	25×7	80	-	0.08	0.025	25×13	100	-
30	0.87	0.012	30×7	92	-	0.1	0.019	30×13	100	-
35	8.53	0.009	35×7	88	-	0.15	0.015	35×13	100	-
40	100.99	0.007	40×7	92	-	0.46	0.011	40×13	100	-
45	403.52	0.006	45×7	80	16	1.09	0.009	45×14	100	-

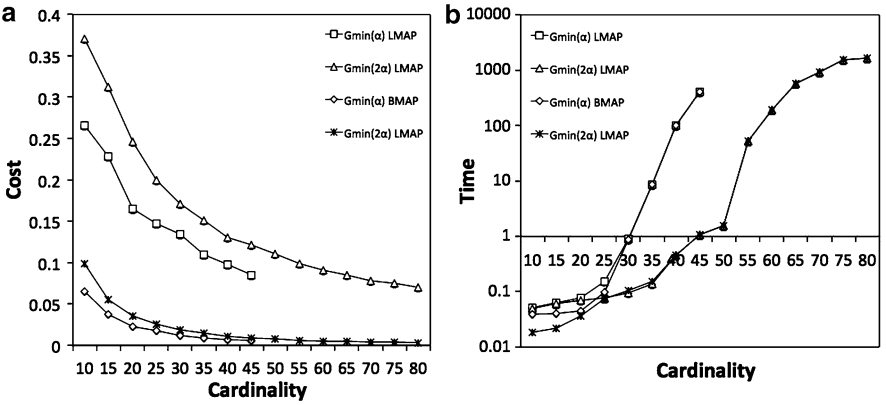

Fig. 5 Comparison of the cost **(a)** and computational time **(b)** for MAPs with linear sum and linear bottleneck objective functions for group (ii) and (iii)

$n = 45$, only 4 (16%) could not be solved in 1 h. Out of the 21 remaining instances, 20 instances contained a clique, and only 1 did not have a clique. The behavior of the average cost values for the problems solved in this group are depicted in Fig. 5.

Finally, the third group, (iii), includes instances for which the cardinality of the problem prevents the α-set from being solved within 1 h. Thus, for this set, only the 2α-set is used. The instances of this group were solved with the parameter values $n = 50, 55, \ldots, 80$ and $d = 3$. Tables 5 and 6 summarize the corresponding results. When the size of the problem $n \geq 55$, some instances of problems become impossible to solve within 1 h time limit. The average cost for the instances that are solved keeps the usual trend and converges to 0 as n grows. The largest problems attempted to be solved in this group are MAPs with $n = 80$. Out of 25 instances of this size, only four could be solved within 1 h. Figure 5(a) the average values of solution cost and computational time for the instances of both linear sum and linear bottleneck MAPs. Note that as the size of the problem increases, the reduction in the size of problem achieved from using α-set or 2α-set becomes significantly larger.

Table 5 Computational time and cost for the first clique found in $\mathcal{G}^*(2\alpha)$ in random MAPs with linear sum objective functions for instances in group (iii)

	$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{\min}(2\alpha)}$	$Z_{\mathcal{G}_{\min}(2\alpha)}$	$\lvert V\rvert$	$\exists$ CLQ	Timeout
50	1.56	0.11	50×14	100	–
55	52.29	0.099	55×14	96	4
60	189.9	0.091	60×14	92	8
65	568.9	0.085	65×14	96	4
70	919.79	0.078	70×14	64	36
75	1556.89	0.075	75×14	40	60
80	1641.26	0.07	80×14	16	84

Table 6 Computational time and cost for the first clique found in $\mathcal{G}^*(2\alpha)$ in random MAPs with linear bottleneck objective functions for instances in group (iii)

	$\mathcal{G}^*_{\min}(2\alpha)$				
n	$T_{\mathcal{G}_{\min}(2\alpha)}$	$W_{\mathcal{G}_{\min}(2\alpha)}$	$\lvert V\rvert$	$\exists$ CLQ	Timeout
50	1.56	0.008	50×14	100	–
55	52.19	0.006	55×14	96	4
60	190.6	0.005	60×14	92	8
65	566.71	0.005	65×14	96	4
70	920.44	0.004	70×14	64	36
75	1552.74	0.004	75×14	40	60
80	1631.89	0.003	80×14	16	84

For instance, in MAP with $n = 80$ and $d = 3$, the 2α-set has 80×14 nodes, while the complete index graph will have 80×80^2 nodes.

3.3 Random MAPs of Large Dimensionality

The second set of problem instances includes MAPs that are solved by the heuristic method explained in Sect. 2.2. Problems in this set have the cardinality $n = 2, \ldots, 5$ and dimensionality in the range $d = 2, \ldots, \bar{d}_n$, where $\bar{d}_n$ is the largest value for d for which an MAP with cardinality n can be solved within 1 h using the heuristic method. For each pair of (n, d), 25 instances of MAP with cost coefficients randomly drawn from the uniform $U[0, 1]$ distribution are generated. Generated instances are then solved by the modified FINDCLIQUE for the optimal clique (when possible) and the optimal costs are compared with the costs obtained from the heuristic method. The result of the heuristic method for instances with $n = 2$ is optimal, and the heuristic checks all the 2^{d-1} solutions of the MAP. Thus, using the modified FINDCLIQUE to find the optimum clique is not necessary.

Figure 6 demonstrates the cost convergence in instances with $n = 2, 3, 4, 5$ for both linear sum and linear bottleneck MAPs. Figure 6(a) demonstrates the cost convergence in MAPs with $n = 2$ and $d = 2, \ldots, 27$. Recall that due to Remark 2, for cases with $n = 2$ the heuristic provides the optimal solution. The heuristic method provides high-quality solutions that are consistently converging to

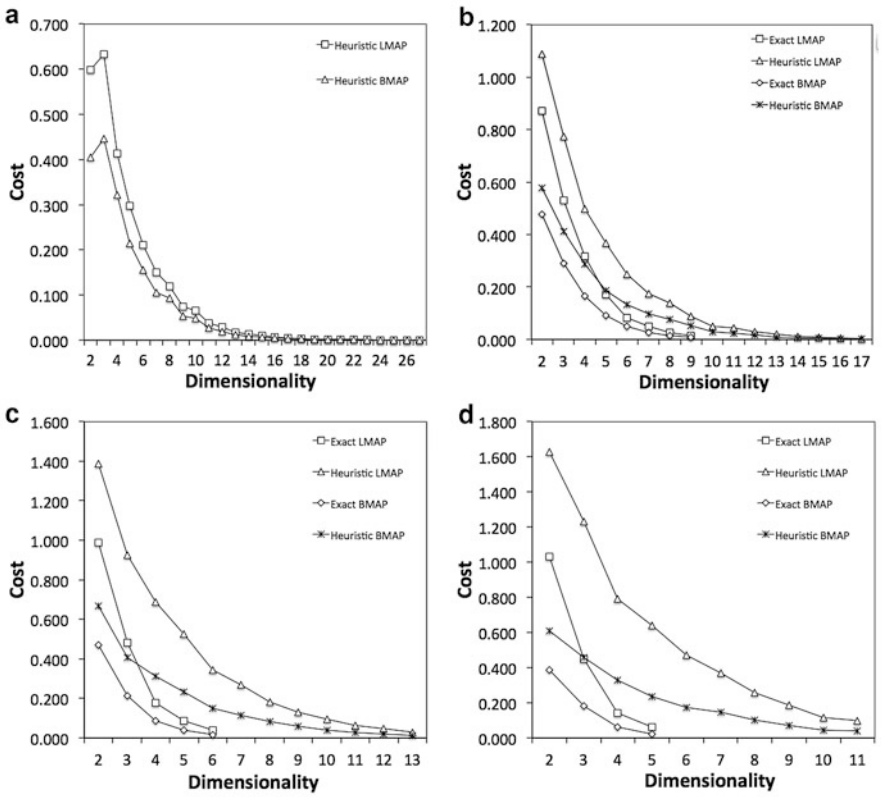

Fig. 6 Comparison of the cost obtained from the heuristic method with the optimum cost in MAPs with linear sum and linear bottleneck objective functions with **(a)** $n = 2$, **(b)** $n = 3$, **(c)** $n = 4$, and **(d)** $n = 5$

the optimal solution for all cases and the average value of the obtained costs from the heuristics approaches 0. Memory limitations, as opposed to computational time, were the restrictive factor for solving larger instances as the computational time for the problems of this set never exceeded 700 s.

Figure 7 demonstrates the computational time for the optimal method as well as the heuristic method in instances with $n = 2, 3, 4, 5$ for both linear sum and linear bottleneck MAPs. The computational time has an exponential trend as the number of solutions for the MAP, or the number of solutions checked by the heuristic grow in an exponential manner. However, the heuristic method is able to find high-quality solutions in significantly shorter time.

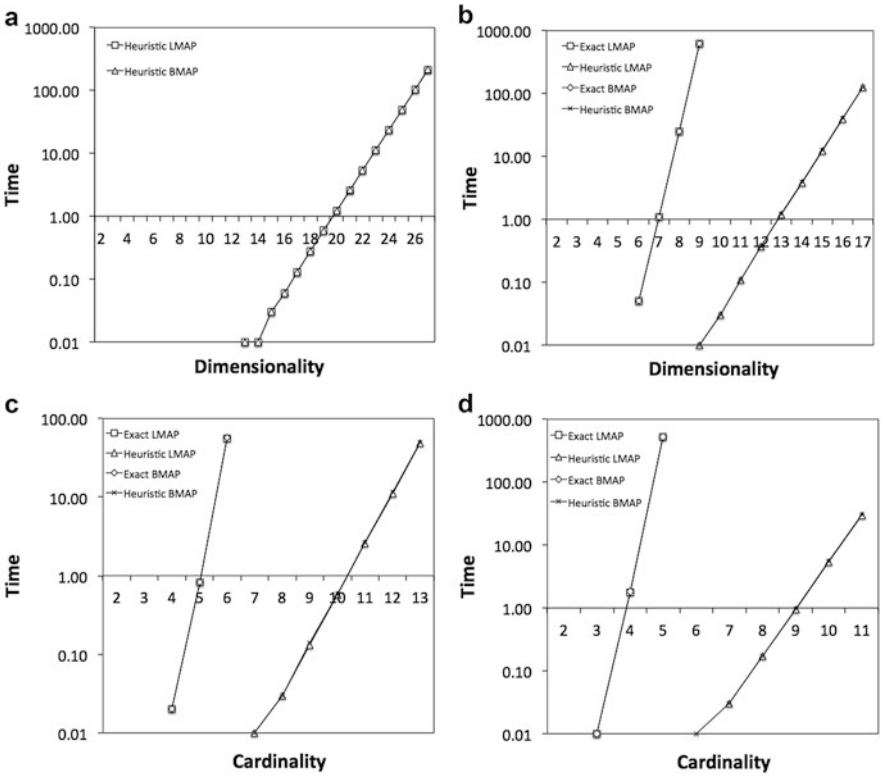

Fig. 7 Comparison of the computational time in logarithmic scale needed for the optimal method and the heuristic method in MAPs with linear sum and linear bottleneck objective functions with **(a)** $n = 2$, **(b)** $n = 3$, **(d)** $n = 4$, and **(d)** $n = 5$

4 Conclusions

In this paper, randomized MAPs that correspond to hypergraph matching problems were studied. Two different methods were provided to obtain guaranteed high-quality solutions for MAPs with linear sum or linear bottleneck cost function and fixed dimensionality and fixed cardinality. The computational results demonstrated that the proposed methods provide a tight upper bound on the value of the optimal cost for MAPs in randomized problems. With the first method, problem instances with $d = 3$ and n as large as 80 are solved. The heuristic provided for problems with fixed cardinality can provide high-quality solutions to problems with large dimensionality in a relatively short time. The limiting factor for the heuristic method is the memory consumption. The structure of the proposed methods makes them suitable for parallel computing. As an extension, the performance of the proposed heuristic in a parallel system will be studied.

References

1. Abdullah, S., Burke, E.K., Mccollum, B.: Using a randomised iterative improvement algorithm with composite neigjhbourhood structures for university course timetabling. In: The Proceedings of the 6th Metaheuristic International Conference [MIC05, pp. 22–26. Book (2005)
2. Bekker, H., Braad, E.P., Goldengorin, B.: Using bipartite and multidimensional matching to select the roots of a system of polynomial equations. In: ICCSA (4), pp. 397–406 (2005)
3. Burkard, R.E.: Selected topics on assignment problems. Discrete Appl. Math. **123**(1–3), 257–302 (2002)
4. Burkard, R.E., Çela, E.: Linear assignment problems and extensions. In: Du, D.-Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization, Supplement Volume A, pp. 75–149. Kluwer Academic Publishers, Dordrecht, (1999)
5. Carter, M.W., Laporte, G.: Recent developments in practical course timetabling. In: Burke, E., Carter, M. (eds.) Practice And Theory Of Automated Timetabling Ii, 2nd International Conference on the Practice and Theory of Automated Timetabling (Patat 97), Toronto, Canada, 20–22 Aug 1997. Lecture Notes In Computer Science, vol. 1408 , pp. 3–19. Springer-Verlag Berlin, Heidelberger Platz 3, D-14197 Berlin, Germany (1998)
6. Dutta, A., Tsiotras, P.: A greedy random adaptive search procedure for optimal scheduling of p2p satellite refueling. In: AAS/AIAA Space Flight Mechanics Meeting, pp. 07–150 (2007)
7. Grundel, D., Krokhmal, P., Oliveira, C., Pardalos, P.: On the number of local minima in the multidimensional assignment problem. J. Combin. Opt. **13**(1), 1–18 (2007)
8. Grünert, T., Irnich, S., Zimmermann, H., Schneider, M., Wulfhorst, B.: Finding all k-cliques in k-partite graphs, an application in textile engineering. Comput. Oper. Res. **29**(1), 13–31 (2002)
9. Krokhmal, P., Grundel, D., Pardalos, P.: Asymptotic behavior of the expected optimal value of the multidimensional assignment problem. Mathem. Prog. **109**(2–3), 525–551 (2007)
10. Krokhmal, P.A., Pardalos, P.M.: Limiting optimal values and convergence rates in some combinatorial optimization problems on hypergraph matchings. Submitted for publication, 3131 Seamans Center, Iowa City, IA 52242, USA, (2011)
11. Kuhn, H.W.: The hungarian method for the assignment problem. Nav. Res. Logist. Quar. **2**(1–2), 83–87 (1955)
12. Pierskalla, W.: The multidimensional assignment problem. Oper. Res. **16**(2), 422–431 (1968)
13. Poore, A.B.: Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. Comput. Opt. Appl. **3**(1), 27–54 (1994)
14. Pusztaszeri, J.F., Rensing, P.E., Liebling, T.M.: Tracking elementary particles near their primary vertex: A combinatorial approach. J. Global Optim. **9**(1), 41–64 (1996) 3rd Workshop on Global Optimization, SZEGED, Hungary, Dec 1995.
15. Urban, T.L., Russell, R.A.: Scheduling sports competitions on multiple venues. European J. Oper. Res. **148**(2), 302–311 (2003)

On Some Special Network Flow Problems: The Shortest Path Tour Problems

Paola Festa

Abstract This paper describes and studies the shortest path tour problems, special network flow problems recently proposed in the literature that have originated from applications in combinatorial optimization problems with precedence constraints to be satisfied, but have found their way into numerous practical applications, such as for example in warehouse management and control in robot motion planning. Several new variants belonging to the shortest path tour problems family are considered and the relationship between them and special facility location problems is examined. Finally, future directions in shortest path tour problems research are discussed in the last section.

Keywords Shortest path tour problems • Network flow problems • Combinatorial optimization

1 Introduction

Shortest path tour problems (SPTPs) are special network flow problems recently proposed in the literature [14]. Given a weighted directed graph G, the classical SPTP consists of finding a shortest path from a given origin node to a given destination node in the graph G with the constraint that the optimal path should successively pass through at least one node from given node mutually independent subsets $T_1, T_2, \ldots, T_N$, where $\bigcap_{k=1}^{N} T_k = \emptyset$. In more detail, P starts at the origin node (that without loss of generality can be assumed in T_1), moves to some node in T_2 (possibly through some intermediate nodes that are not in T_2), then moves to some node in T_3 (possibly through some intermediate nodes that are not in

P. Festa (✉)
Department of Mathematics and Applications, University of Napoli FEDERICO II, Compl. MSA, Via Cintia, 80126 Napoli, Italy
e-mail: paola.festa@unina.it

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_13,

T_3, but may be in T_1 and/or in T_2), etc., then finally it moves to the destination node (possibly through some intermediate nodes not equal to the destination, which without loss of generality can be assumed in T_N).

The SPTP and the idea behind it were given in 2005 as Exercise 2.9 in Bertsekas's *Dynamic Programming and Optimal Control* book [3], where it is asked to formulate it as a dynamic programming problem. Very recently, in [14] it has been proved that the SPTP belongs to the complexity class **P**. This problem has several practical applications, such as, for example, in warehouse management or control of robot motions. In both cases, there are precedence constraints to be satisfied. In the first case, assume that an order arrives for a certain set of N collections of items stored in a warehouse. Then, a vehicle has to collect at least an item of each collection of the order to ship them to the costumers. In control of robot motions, assume that to manufacture workpieces, a robot has to perform at least one operation selected from a set of N types of operations. In this latter case, operations are associated with nodes of a directed graph and the time needed for a tool change is the distance between two nodes.

The remainder of this article is organized as follows. In Sect. 2, the classical SPTP is described and its properties are analyzed. Exact techniques proposed in [14] are also surveyed along with the computational results obtained and analyzed in [14]. In Sect. 3, several new different variants of the classical SPTP are stated and formally described as special facility location problems. Concluding remarks and future directions in SPTPs research are discussed in the last section.

2 Notation and Problem Description

Throughout this paper, the following notation and definitions will be used.

Let $G = (V, A, C)$ be a directed graph, where

- V is a set of nodes, numbered $1, 2, \ldots, n$.
- $A = \{(i, j) | \, i, j \in V\}$ is a set of m arcs.
- $C : A \mapsto \mathbb{R}^+ \cup \{0\}$ is a function that assigns a nonnegative length c_{ij} to each arc $(i, j) \in A$.
- For each node $i \in V$, let $FS(i) = \{j \in V \mid (i, j) \in A\}$ and $BS(i) = \{j \in V \mid (j, i) \in A\}$ be the *forward star* and *backward star* of node i, respectively.
- A *simple path* $P = \{i_1, i_2, \ldots, i_k\}$ is a walk without any repetition of nodes.
- The length $L(P)$ of any path P is defined as the sum of lengths of the arcs connecting consecutive nodes in the path.

Then, the SPTP can be stated as follows:

Definition 1. The SPTP consists of finding a shortest path from a given origin node $s \in V$ to a given destination node $d \in V$ in the graph G with the constraint that the optimal path P should successively pass through at least one node from given node subsets $T_1, T_2, \ldots, T_N$, where $\bigcap_{k=1}^{N} T_k = \emptyset$.

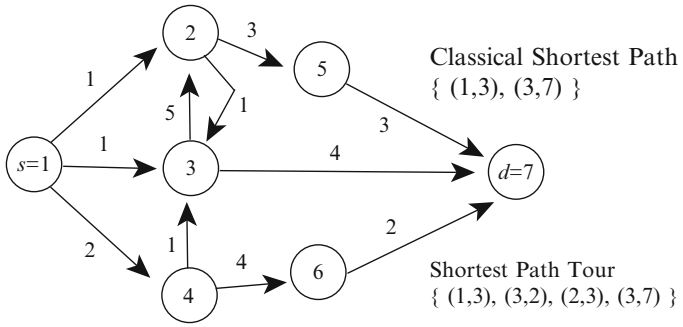

Fig. 1 A SPTP instance on a small graph G

Let us consider the small graph $G = (V, A, C)$ depicted in Fig. 1, where $V = \{s = 1, 2, \ldots, 7 = d\}$. It is easy to see that $P = \{1, 3, 7\}$ is the shortest path from node 1 to node 7 and has length 5. Let us now define on the same small graph G the SPTP instance characterized by $N = 4$ and the following node subsets $T_1 = \{s = 1\}$, $T_2 = \{3\}$, $T_3 = \{2, 4\}$, $T_4 = \{d = 7\}$. The shortest path tour from node 1 to node 7 is the path $P_T = \{1, 3, 2, 3, 7\}$ which has length 11 and is not simple, since it passes twice through node 3.

When dealing with any given optimization problem (such as SPTP), one is usually interested in classifying it according to its complexity in order to be able to design an algorithm that solves the problem of finding the best compromise between solution quality and computational time required to find that solution. In classifying a problem according to its complexity, polynomial-time reductions are helpful. In fact, deciding the complexity class of an optimization problem Pr becomes easy once a polynomial-time reduction to a second problem $\bar{P}r$ is available and the complexity of $\bar{P}r$ is known, as stated in Definitions 2 and 3 and Theorem 1, whose proof is reported in [17] and several technical books, including [18].

Definition 2. A problem Pr is Karp-reducible to a problem $\bar{P}r$ ($Pr <_m \bar{P}r$) if there exists a function f such that

$$x \text{ is a positive instance of } Pr \iff f(x) \text{ is a positive instance of } \bar{P}r; \qquad (1)$$

f is called *Karp reduction function* and an algorithm $\mathcal{A}$ that computes f is called *a Karp reduction algorithm*.

If both $Pr <_m \bar{P}r$ and $\bar{P}r <_m Pr$, Pr and $\bar{P}r$ are Karp-equivalent ($Pr \equiv_m \bar{P}r$).

Definition 3. A problem Pr is polynomially Karp-reducible to a problem $\bar{P}r$ ($Pr <_m^p \bar{P}r$) if there exists a polynomial-time computable function f such that

$$x \text{ is a positive instance of } Pr \iff f(x) \text{ is a positive instance of } \bar{P}r; \qquad (2)$$

f is called *Karp reduction function* and a polynomial-time algorithm $\mathcal{A}$ that computes f is called *a Karp reduction algorithm*.

If both $Pr <_m^p \bar{P}r$ and $\bar{P}r <_m^p Pr$, Pr and $\bar{P}r$ are polynomially Karp-equivalent ($Pr \equiv_m^p \bar{P}r$).

Theorem 1. *Let Pr and $\bar{P}r$ be any two optimization problems such that $Pr <_m^p \bar{P}r$, then $\bar{P}r$ in the complexity class* **P** *of polynomially solvable problems implies $Pr \in$* **P**.

Hence, Theorem 1 guarantees that problem Pr and problem $\bar{P}r$ belong to the same complexity class, or in other words problem Pr is no harder than problem $\bar{P}r$. In [14], the author used a polynomial-time reduction and theoretical result of Theorem 1 to prove that SPTP belongs to the complexity class **P**, since it reduces to a single source—single destination shortest path problem (SPP). In the following, for sake of clarity, we report this complexity results stated and proved in [14].

Theorem 2. *SPTP $<_m^p$ SPP, then SPTP $\in$* **P**.

Proof. To prove the thesis, a polynomial-time reduction algorithm must be found that transforms any SPTP instance into a single source–single destination SPP instance and vice versa.

It is trivial to show that any SPP instance $< G = (V, A, C), s, d >$ can be polynomially transformed in the SPTP instance $< G = (V, A, C), s, d, N, \{T_k\}_{k=1,\ldots,N} >$, where $N = 2$ and $T_1 = \{s\}$, $T_2 = V \setminus \{s\}$.

Conversely, there exists a polynomial-time reduction algorithm that transforms any SPTP instance $< G = (V, A, C), s, d, N, \{T_k\}_{k=1,\ldots,N} >$ into a single source–single destination SPP instance $< G' = (V', A', C'), s, d' = d + (N-1) \cdot n >$, where G' is a multistage graph with N stages, each replicating G. Figure 2 depicts the pseudo-code of the reduction algorithm `SPTPReduction` that performs the following operations.

(i) Line 1—$V' := \{1, 2, \ldots, N \cdot n\}$; $A' := \emptyset$:
The set of nodes V' and the set of arcs A' of the multistage graph G' are initialized. The set V' has n nodes for each stage $k \in \{1, \ldots, N\}$; the set A' is initially empty.

(ii) Loop for in lines 2–12: the stages $1, \ldots, N-1$ are constructed.
At each iteration, an arc (a, b) is added to A'. In particular, for each stage $k \in \{1, \ldots, N-1\}$, for each node $v \in \{1, \ldots, n\}$, and for each adjacent node $w \in FS(v)$, $(a, b) = (v + (k-1) \cdot n, w + k \cdot n)$ with length c_{vw}, if $w \in T_{k+1}$; $(a, b) = (v + (k-1) \cdot n, w + (k-1) \cdot n)$ with length c_{vw}, otherwise.

(iii) Loop for in lines 13–17: the stage N is completed.
At each iteration, for each node $v \in \{1, \ldots, n\}$ and for each adjacent node $w \in FS(v)$, an arc (a, b) is added to A' connecting node $a = v + (N-1) \cdot n$ to node $b = w + (N-1) \cdot n$ and having length c_{vw}.

It is easy to see that $|A'| = N \cdot m$ and therefore the computational complexity of `SPTPReduction` is $O(N \cdot m)$. Note that, since $\bigcap_{k=1}^{N} T_k = \emptyset$, it results that $N \leq n$, and therefore, the worst case computational complexity is $O(n \cdot m)$.

```
algorithm SPTPReduction(V,A,C,N,(T_k)_{k=1,2,...,N})
1 V' := {1, 2, ..., N * n}; A' := ∅;
2 for k = 1 to N − 1 →
3     for v = 1 to n →
4         for each w ∈ FS(v) →
5             if (w ∈ T_{k+1}) then
6                 add(A',(v + (k − 1) * n, w + k * n),c_{vw});
7             else
8                 add(A',(v + (k − 1) * n, w + (k − 1) * n),c_{vw});
9             endif
10        endfor
11    endfor
12 endfor
13 for v = 1 to n →
14    for each w ∈ FS(v) →
15        add(A',(v + (N − 1) * n, w + (N − 1) * n),c_{vw});
16    endfor
17 endfor
18 return (V',A',C');
end SPTPReduction
```

Fig. 2 Pseudo-code of a polynomial reduction algorithm

Figure 3 depicts the multistage graph G' corresponding to the SPTP instance on the small graph G of Fig. 1. Note that, $|V'| = N \cdot n = 4 \cdot 7 = 28$, $|A'| = N \cdot m = 4 \cdot 11 = 44$, $s = 1$, $d' = d + (N-1) \cdot n = 7 + 3 \cdot 7 = 28$.

We now claim that in G', as constructed above, there is a path P' from s to d' of length K if and only if in G there is a path tour P_T from s to d of length K. Suppose that in G' there is a path P' of length K from s to d'. Because P' connects $s \in T_1$ to $d' \in T_N$, for each $k = 1, \ldots, N-1$ there must be in P' necessarily at least one arc connecting two nodes in consecutive stages k and $k+1$. Therefore, it follows that P' consists of at least one node in each of the N stages, so corresponding to a path tour P_T of length K in G that successively passes through at least one node from the given node subsets $T_1, T_2, \ldots, T_N$.

Conversely, suppose that in G there is a path tour P_T of length K that successively passes through at least one node from the given node subsets $T_1, T_2, \ldots, T_N$. Then, by construction, for each arc in P_T connecting in G two nodes belonging to consecutive subsets, there exists in the simple path P' an arc connecting in G' two nodes belonging to consecutive stages, till finally moving to d' in the last stage N.

□

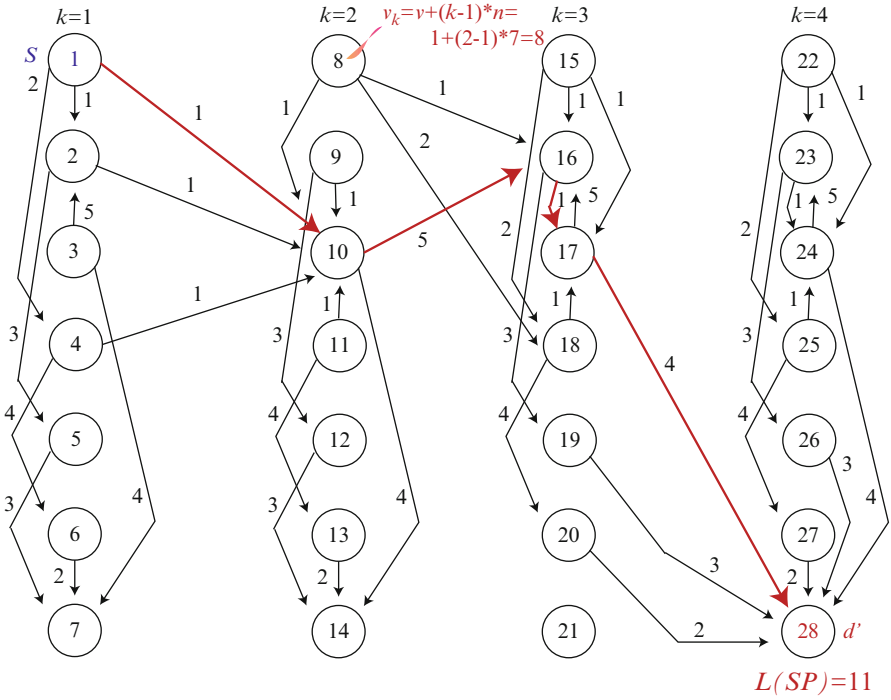

Fig. 3 The multistage graph G' corresponding to the SPTP instance on G of Fig. 1

2.1 Several Alternative Algorithms for the SPTP

Several alternative techniques have been designed in [14] to exactly solve the SPTP as defined in Sect. 1. Once obtained in polynomial-time the multistage graph G' by applying the algorithm `SPTPReduction`, any shortest path algorithm can be applied to solve the resulting SPP.

Classical SPPs are among the most studied combinatorial problems that arise as subproblems when solving many optimization problems. Exhaustive surveys of the most interesting and efficient shortest path algorithms, important for their computational time complexity or for their practical efficiency, can be found among others in [1, 7–12, 15, 16, 19]. Although the huge number of state-of-the-art algorithms for the SPP, there does not exist a *best* method that outperforms all the others. In fact, recent research lines tend to develop techniques designed *ad hoc* for solving special structured SPPs: either a special network topology or a special cost structure.

In [14], the following algorithms have been designed and tested:

- A dynamic programming algorithm, as suggested in [3].
- A Dijkstra-like algorithm [13] that uses a binary heap to store the nodes with temporary labels.

- Several Auction-like algorithms [2, 4]: a *forward*, a *backward*, and a *combined for/backward* version.

To describe the dynamic programming approach a slight different notation to represent an *expanded graph* $G' = (V', A', C')$ has been used.

For each node $i \in V$, V' contains $N + 1$ nodes $(i, 0)$, $(i, 1)$, ..., (i, N). The meaning of being in node (i, k), $k = 1, 2, \ldots, N$, is that we are at node i and have already successively visited the sets $T_1, \ldots, T_k$, but not yet the sets $T_{k+1}, \ldots, T_N$. The meaning of being in node $(i, 0)$ is that we are at node i and have not yet visited any node in the set T_1. For each arc $(i, j) \in A$ and for each $k = 0, 1, \ldots, N - 1$, we introduce in A' an arc from (i, k) to (j, k), if $j \notin T_{k+1}$ or an arc from (i, k) to $(j, k + 1)$, if $j \in T_{k+1}$. Moreover, for each arc $(i, j) \in A$ we introduce in A' an arc from (i, N) to (j, N). Once obtained the expanded graph G', the SPTP is equivalent to find a shortest path from $(s, 0)$ to (d, N).

Let $D^{r+1}(i, k)$, $k = 0, 1, \ldots, N$, be the shortest distance from (i, k) to the destination node (d, N) using r arcs or less.

For $k = 0, 1, \ldots, N - 1$ the DP iteration is the following:

$$D^{r+1}(i,k) = \min_{(i,j)\in A} \left\{ \min_{j\notin T_{k+1}} \{c_{ij} + D^r(j,k)\}, \min_{j\in T_{k+1}} \{c_{ij} + D^r(j,k+1)\} \right\}$$

$$D^{r+1}(i,N) = \begin{cases} \min\limits_{(i,j)\in A} \{c_{ij} + D^r(j,N)\}, & \text{if } i \neq d; \\ 0, & \text{if } i = d. \end{cases} \tag{3}$$

Initial conditions are the following:

$$D^0(i,k) = \begin{cases} \infty, & \text{if } (i,k) \neq (d,N); \\ 0, & \text{if } (i,k) = (d,N). \end{cases}$$

In [14] it has been proved that the dynamic programming algorithm implementing the above DP iteration is correct and terminates in a finite number of iterations, as stated in the following theorem.

Theorem 3. *An algorithm implementing the above DP iteration terminates after a finite number of iterations with an optimal solution and its computational complexity is* $O(N^2 \cdot n \cdot m)$, *and* $O(n^3 \cdot m)$ *in the worst case.*

2.2 Experimental Results

In this subsection, we report on some computational experiments carried in [14] to determine which algorithm among those proposed seems to be more effective to solve the classical SPTP.

The following algorithms have been designed and implemented:

(a) `Dijkstra` forward with binary heap
(b) Standard `Auction forward`
(c) Standard `Auction backward`
(d) Combined for/backward `Auction fb`
(e) `DP Dijkstra`, DP with Dijkstra as initialization phase
(f) `DP Auction`, DP with Auction as initialization phase
(g) `DP Auction back`, DP with Auction backward as initialization phase
(h) `DP Dijkstra back`, DP with Dijkstra backward as initialization phase

Since the algorithm simply implementing DP iteration (3) was too time consuming, in [14] a slightly different variant was proposed that first calculates $D(i, N)$ using a standard shortest path computation (`DP Dijkstra`, `DP Auction`, `DP Dijkstra`, `DP Auction back`, `DP Dijkstra back`).

The objectives of the computational study were to compare the running times achieved by several alternative algorithms as a function of the parameter N when applied to solve SPTP instances pseudorandomly generated and characterized by several different network topologies, with different densities and number of nodes. The arc lengths have been pseudorandomly generated as integers in the range from 0 to 10000 and two nodes have been randomly chosen to be the source node s and the destination node d, respectively. Moreover, the following graph families have been considered: (1) complete graphs with $n \in \{60, 100\}$; (2) square grids with $n = 10 \times 10$ and rectangular grids with $n = 25 \times 6$; (3) random graphs with $n = 150$ and $m \in \{4 \times n, 8 \times n\}$.

For each problem family, ten different instances have been generated for each possible value of $N \in \{10\%n, 30\%n, 50\%n, 70\%n\}$ and the mean time (in second) required to find an optimal solution has been stored and plotted in Figs. 4–9.

Looking at the results obtained and analyzed in [14], Dijkstra's algorithm outperforms all the competitors.

3 Several New Different Variants of the Classical SPTP

In this section, several new different variants of the classical SPTP are stated and new results are presented about their reduction to special facility location problems.

3.1 A Special Non-metric Multilevel Uncapacitated Facility Location Problem

In the 1-level uncapacitated facility location problem (1-UFLP), a set of clients and a set of facilities are given and the target is to find a subset of facilities to be opened such that all the clients are served by the open facilities while minimizing the total cost of opening facilities and serving clients. In the more general l-level

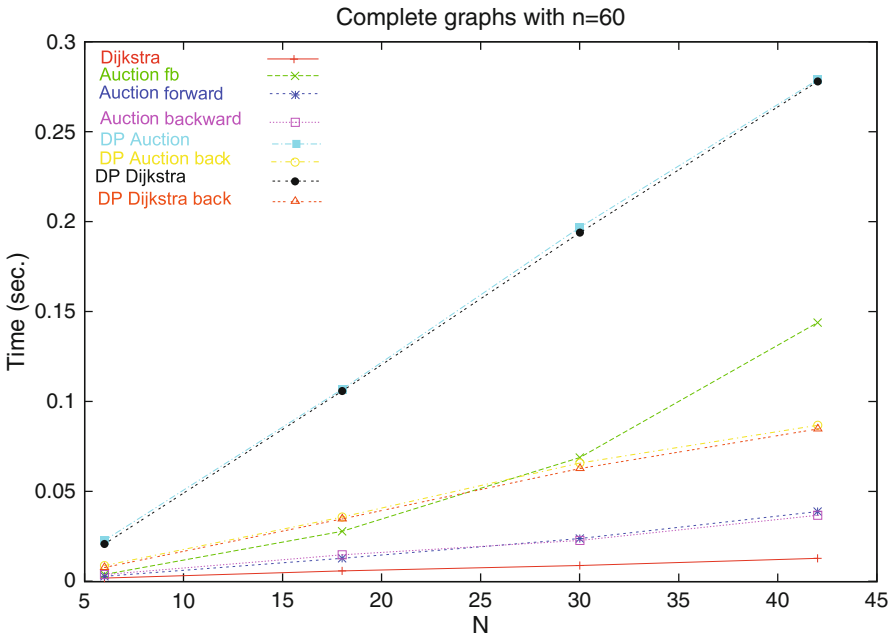

Fig. 4 Complete graphs with $n = 60$: ten different instances have been generated for each possible value of $N \in \{10\%n, 30\%n, 50\%n, 70\%n\}$. For each algorithm and for each instance, the mean running time (over ten trials) required to find an optimal solution has been stored and plotted as a function of the parameter N of node subsets

uncapacitated facility location problem (l-UFLP), the demands must be routed among facilities in a hierarchical order, i.e., from the highest level (for example, the factories) down to the lowest (for example, the retailers), before reaching the clients. More formally, the l-UFLP can be stated as follows.

Given

- The set of clients D
- l-level sets of sites $\mathcal{F}_1, \ldots, \mathcal{F}_l$, i.e., $\mathcal{F}_k$, $k = 1, \ldots, l$, is the set of sites where facilities may be located on level k
- $\mathcal{F} = \bigcup_{k=1}^{l} \mathcal{F}_k$
- The cost $f_{i_k} > 0$ of setting up facility at site $i_k \in \mathcal{F}_k$, $k = 1, \ldots, l$
- A function $C : D \cup \mathcal{F} \times D \cup \mathcal{F} \to \mathbb{R}^+ \cup \{0\}$ that assigns a nonnegative cost $c_{ab} \geq 0$ of connecting $a, b \in D \cup \mathcal{F}$ (the adjective *non-metric* stands because any assumptions are made on the connecting costs, such as symmetry and/or triangle inequality)

each client $j \in D$ must be served by exactly one *open path* $P = \{i_1, \ldots, i_l\} \in \mathcal{P} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_l$ of l facilities with exactly one from each of the l-levels, where a path P is open if and only if every facility on P is open. The total service cost Γ_{jP} incurred by assigning a client $j \in D$ to an open path $P = \{i_1, \ldots, i_l\} \in \mathcal{P}$ is the

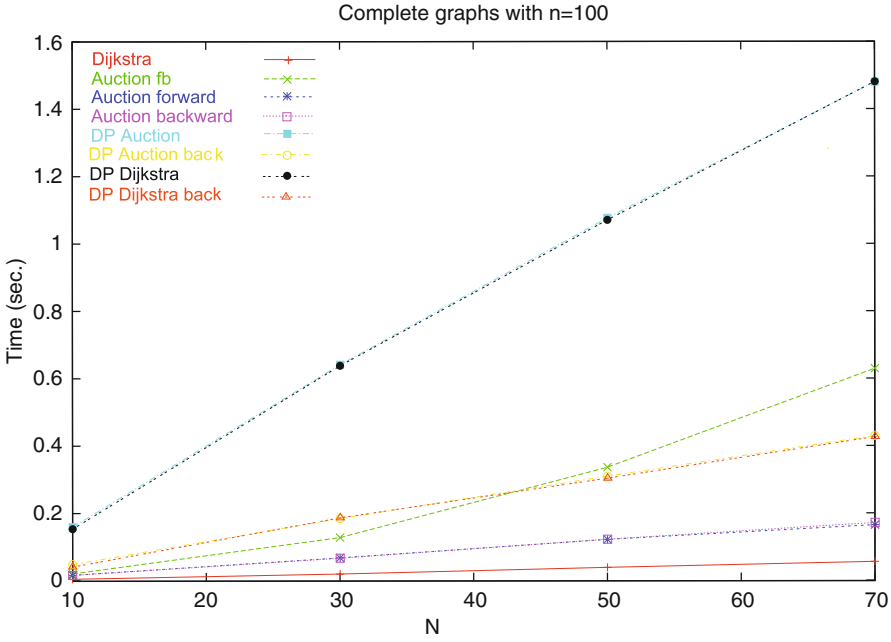

Fig. 5 Complete graphs with $n = 100$: ten different instances have been generated for each possible value of $N \in \{10\%n, 30\%n, 50\%n, 70\%n\}$. For each algorithm and for each instance, the mean running time (over ten trials) required to find an optimal solution has been stored and plotted as a function of the parameter N of node subsets

total connection cost given by

$$\Gamma_{jP} = \sum_{k=1}^{l-1} c_{i_k i_{k+1}} + c_{i_l j}.$$

Given a l-UFLP instance $< D \cup \mathcal{F}, C, f(\cdot) >$, the objective is to open a subset of facilities such that each client is assigned to an open path and the total cost is minimized, i.e., to choose $\emptyset \neq S_k \subset \mathcal{F}_k, k = 1, \ldots, l$ such that

$$\sum_{j \in D} \min_{P \in S_1 \times \cdots \times S_l} \Gamma_{jP} + \sum_{k=1}^{l} \sum_{i_k \in S_k} f_{i_k}.$$

is minimized.

Let us now consider a special SPTP hereafter called SPTP-1 and defined as follows:

Definition 4. The SPTP-1 consists of finding a shortest path from a given origin node $s \in V$ to a given destination node $d \in V$ in the graph G with the constraint

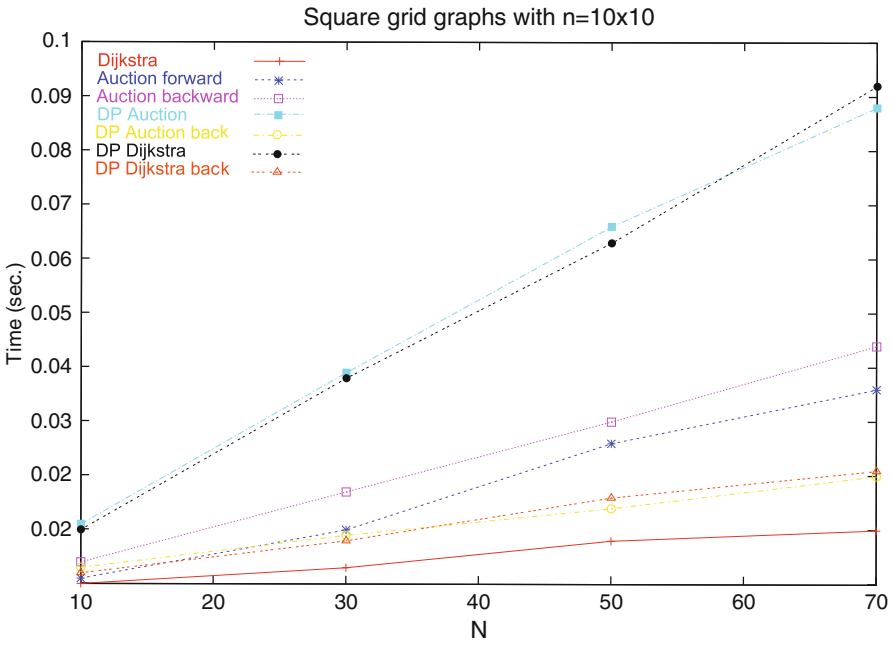

Fig. 6 Square grids with $n = 10 \times 10$: ten different instances have been generated for each possible value of $N \in \{10\%n, 30\%n, 50\%n, 70\%n\}$. For each algorithm and for each instance, the mean running time (over ten trials) required to find an optimal solution has been stored and plotted as a function of the parameter N of node subsets

that the optimal path P should successively pass through exactly and exclusively one node from given node subsets $T_1, T_2, \ldots, T_N$, where $\bigcap_{k=1}^{N} T_k = \emptyset$.

As stated and proved in Theorem 4, the SPTP-1 is polynomially Karp-reducible to a special l-UFLP, where

- $|D| = 1$
- $f_{i_k} = 1$ for each site $i_k \in \mathcal{F}_k, k = 1, \ldots, l$

Let us call this special location problem l-UFLP. It holds the following result.

Theorem 4. *SPTP-1* $<_m^p$ *l-UFLP.*

Proof. To prove the thesis, a polynomial-time reduction algorithm must be found that transforms any SPTP-1 instance into a l-UFLP instance and vice versa.

Let us consider any SPTP-1 instance $< G = (V, A, C), s, d, N, \{T_k\}_{k=1,\ldots,N} >$. It is easy to see that there exists a polynomial-time reduction algorithm that transforms it into the l-UFLP instance $< D \cup \mathcal{F}, C, f(\cdot) >$, where

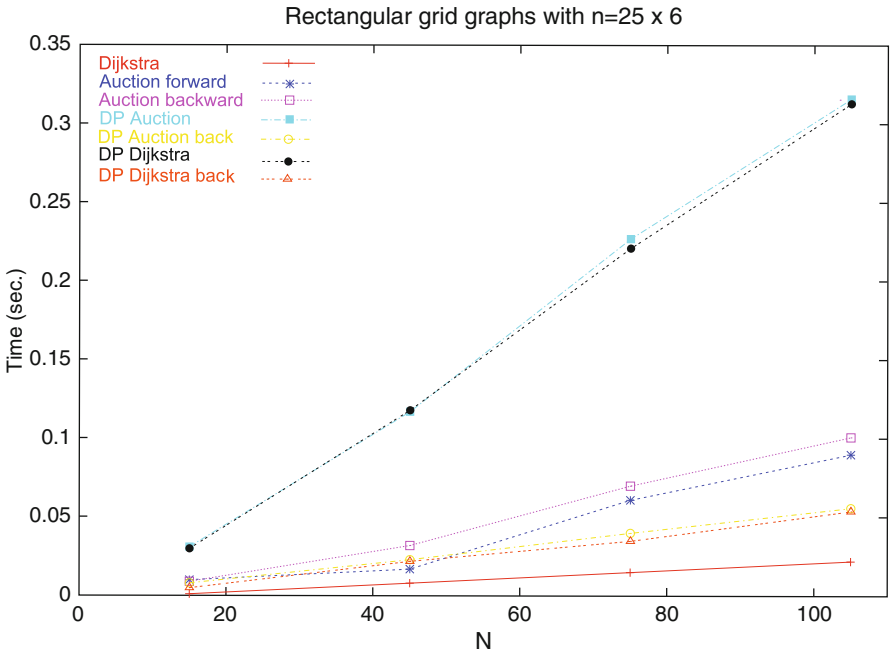

Fig. 7 Rectangular grids with $n = 25 \times 6$: ten different instances have been generated for each possible value of $N \in \{10\%n, 30\%n, 50\%n, 70\%n\}$. For each algorithm and for each instance, the mean running time (over ten trials) required to find an optimal solution has been stored and plotted as a function of the parameter N of node subsets

- $D = \{d\}$.
- $l = N - 1$.
- $\mathcal{F}_k = T_k$ for $k = 2, \ldots, N-1$, $\mathcal{F}_1 = \{s\}$, and $\mathcal{F} = \bigcup_{k=1}^{N-1} \mathcal{F}_k$.
- The connecting cost function $C : D \cup \mathcal{F} \times D \cup \mathcal{F} \rightarrow \mathbb{R}^+ \cup \{0\}$ is the SPTP-1 length function C.

The SPTP-1 instance $< G = (V, A, C), s, d, N, \{T_k\}_{k=1,\ldots,N} >$ admits a solution path tour P_T from s to d of length $l(P_T)$ if and only if l-UFLP instance $< D \cup \mathcal{F}, C, f(\cdot) >$ admits a solution open path $P = \{i_1, \ldots, i_{N-1}\} \in \mathcal{P} = T_1 \times \cdots \times T_{N-1}$ of $N - 1$ facilities with exactly one from each of the $N - 1$ levels and having connection cost $\Gamma_{dP} = l(P_T)$. The total cost of the open path $P = \{i_1, \ldots, i_{N-1}\}$ is $l(P_T) + N - 1$.

Conversely, there exists a polynomial-time reduction algorithm that transforms any l-UFLP instance $< D \cup \mathcal{F}, C, f(\cdot) >$ into a SPTP-1 instance $< G = (V, A, C), s, d, N, \{T_k\}_{k=1,\ldots,N} >$, where

- $V = D \cup \mathcal{F} \cup \{s\}$.
- $d = j \in D$ (remind that $|D| = 1$, hence $D = \{j\}$).

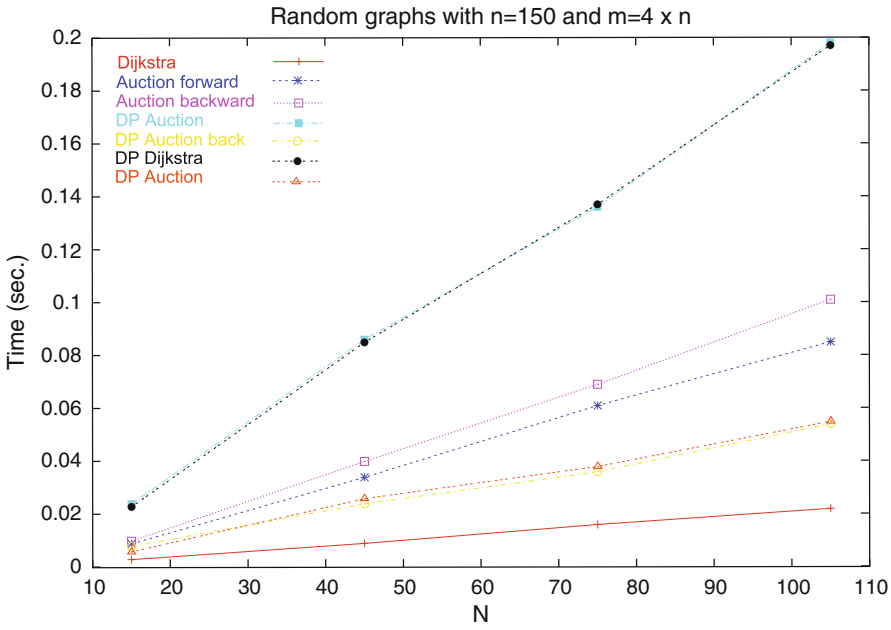

Fig. 8 Random graphs with $n = 150$ and $m = 4 \times n$: ten different instances have been generated for each possible value of $N \in \{10\%n, 30\%n, 50\%n, 70\%n\}$. For each algorithm and for each instance, the mean running time (over ten trials) required to find an optimal solution has been stored and plotted as a function of the parameter N of node subsets

- $N = l + 2$.
- $T_1 = \{s\}$.
- $T_k = \mathcal{F}_{k-1}$ for $k = 2, \ldots, l + 1 = N - 1$ and $T_N = D$.
- The SPTP-1 length function C is the connecting cost function $C : D \cup \mathcal{F} \times D \cup \mathcal{F} \to \mathbb{R}^+ \cup \{0\}$.
- A is made up of all arcs (a, b) corresponding to a connection cost c_{ab} in l-UFLP. Moreover, for each $v_1 \in \mathcal{F}_1$ an arc (s, v_1) is introduced in A with length $c_{sv_1} = 0$.

The l-UFLP instance $< D \cup \mathcal{F}, C, f(\cdot) >$ admits a solution open path $P = \{i_1, \ldots, i_l\} \in \mathcal{P} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_l$ of l facilities with exactly one from each of the l levels and having connection cost Γ_{dP} and total cost $\Gamma_{dP} + l$ if and only if the SPTP-1 instance $< G = (V, A, C), s, d, N, \{T_k\}_{k=1,\ldots,N} >$ admits a solution path tour P_T from s to d of length $l(P_T) = \Gamma_{dP}$. □

Note that, if the connecting costs for the multilevel facility location and the arc lengths for the path tour problem satisfy the triangle inequality, it is straightforward to prove that the *metric* SPTP is equivalent to the *metric* l-UFLP.

As special multilevel uncapacitated facility location problem, SPTP-1 can be formulated as the following integer linear programming problem.

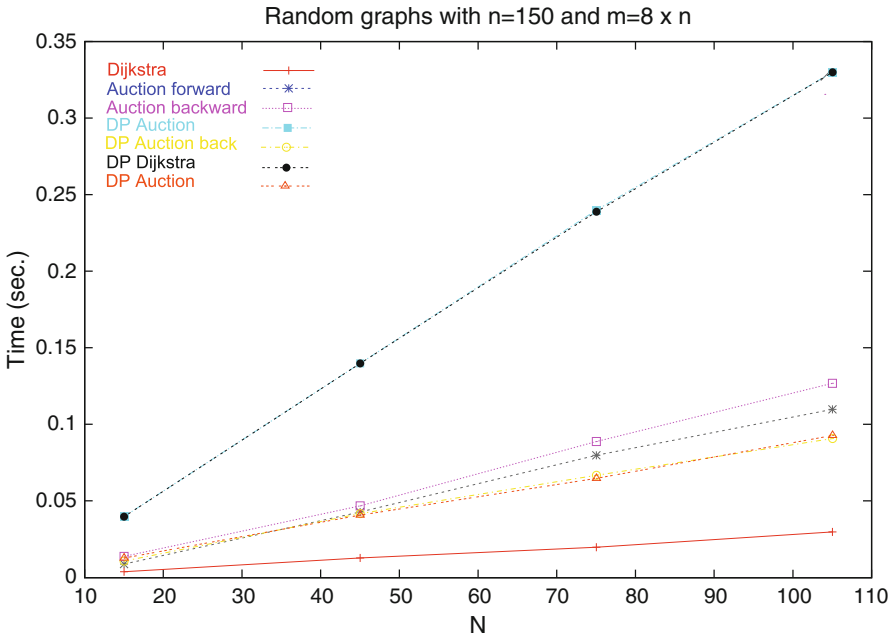

Fig. 9 Random graphs with $n = 150$ and $m = 8 \times n$: ten different instances have been generated for each possible value of $N \in \{10\%n, 30\%n, 50\%n, 70\%n\}$. For each algorithm and for each instance, the mean running time (over ten trials) required to find an optimal solution has been stored and plotted as a function of the parameter N of node subsets

Introducing a Boolean decision variable x_{dP} for each path $P \in \mathcal{P}$ s.t.

$$x_{dP} = \begin{cases} 1, \text{ if node } d \text{ is the terminal node of the path tour } P; \\ 0, \text{ otherwise,} \end{cases}$$

then,

$$\text{(SPTP-1) min} \sum_{P \in \mathcal{P}} \Gamma_{dP}\, x_{dP}$$

$$\text{s.t.}$$

$$\sum_{P \in \mathcal{P}} x_{dP} \geq 1 \tag{5}$$

$$x_{dP} \in \{0, 1\}, \forall\ P \in \mathcal{P} \tag{6}$$

Constraint (5) imposes that the destination node d is terminal node of at least one path tour.

3.2 The Weighted Metric SPTP (W-mSPTP)

The weighted metric SPTP (W-mSPTP) can be stated as follows.

Definition 5. Given a directed graph $G = (V, A, C, W)$, where $W : V \mapsto \mathbb{R}^+$ is a function that assigns a positive weight w_i to each node $i \in V$, and the length function C assigns a nonnegative length c_{ij} to each arc $(i, j) \in A$ such that

- $c_{ij} = c_{ji}$, for each $(i, j) \in A$ (symmetry)
- $c_{ij} \leq c_{ih} + c_{hj}$, for each (i, j), (i, h), $(h, j) \in A$ (triangle inequality)

then, the W-mSPTP consists of finding a minimal cost path (in terms of both total length and total weight of the involved nodes) from a given origin node $s \in V$ to a given destination node $d \in V$ in the graph G with the constraint that the optimal path P should successively pass through at least one node from given node subsets $T_1, T_2, \ldots, T_N$, where $\bigcap_{k=1}^{N} T_k = \emptyset$.

Formally, given a W-mUFLP instance $< G = (V, A, C, W), s, d, N, \{T_k\}_{k=1,\ldots,N} >$, the objective is to find a path $P = \{i_1 = s, \ldots, i_N = d\} \in T_1 \times \cdots \times T_N$ from $s \in T_1$ to $d \in T_N$ corresponding to the minimum total cost, i.e., to choose $\emptyset \neq S_k \subset T_k$, $k = 1, \ldots, N$ such that

$$\min_{P \in S_1 \times \cdots \times S_k} \Lambda(P) + \sum_{k=1}^{N} \sum_{i_k \in S_k} w_{i_k}$$

is minimized, where $\Lambda(P) = \sum_{k=1}^{N-1} c_{i_k i_{k+1}}$.

As stated in Theorem 5, the W-mSPTP is polynomially Karp-reducible to a special l-UFLP, where $|D| = 1$. Let us call this special location problem l-1-UFLP. It holds the following result:

Theorem 5. *W-mSPTP* $<_m^p$ *l-1-UFLP.*

Proof. To prove the thesis, a polynomial-time reduction algorithm must be found that transforms any W-mSPTP instance into a l-1-UFLP instance and vice versa.

Once assimilated the setting up facility costs $f(\cdot)$ with the node weight function $W(\cdot)$, the polynomial reduction can be proved following similar reasonings as for the claim of Theorem 4. The proof is completed by observing that the l-1-UFLP instance $< D \cup \mathcal{F}, C, f(\cdot) >$ admits a solution open path $P = \{i_1, \ldots, i_l\} \in \mathcal{P} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_l$ of l facilities with exactly one from each of the l levels and having connection cost Γ_{dP} and total cost

$$\Gamma_{dP} + \sum_{k=1}^{l} \sum_{i_k \in S_k} f_{i_k}$$

if and only if the W-mSPTP instance $< G = (V, A, C, W), s, d, N, \{T_k\}_{k=1,\ldots,N} >$ admits a solution P from s to d of length $l(P) = \Lambda(P) = \Gamma_{dP}$ and total cost given by

$$\Gamma_{dP} + \sum_{k=1}^{N} \sum_{i_k \in S_k} w_{i_k}. \qquad \square$$

3.3 The Weighted Metric 1-q-SPTP (W-1-q-mSPTP)

Let us consider a further variant of the problem: the weighted metric 1-q-SPTP (W-1-q-mSPTP) stated as follows:

Definition 6. Given a directed graph $G = (V, A, C, W)$, where $W : V \mapsto \mathbb{R}^+$ is a function that assigns a positive weight w_i to each node $i \in V$, and the length function C assigns a nonnegative length c_{ij} to each arc $(i, j) \in A$ such that

- $c_{ij} = c_{ji}$, for each $(i, j) \in A$ (symmetry)
- $c_{ij} \leq c_{ih} + c_{hj}$, for each (i, j), (i, h), $(h, j) \in A$ (triangle inequality)

then, the W-1-q-mSPTP consists of finding a minimal cost path (in terms of both total length and total weight of the involved nodes) from a given origin node $s \in V$ to each destination node $d_r \in D \subseteq T_N$ in the graph G with the constraint that each corresponding optimal path P_{d_r} should successively pass through at least one node from given node subsets $T_1, T_2, \ldots, T_N$, where $\bigcap_{k=1}^{N} T_k = \emptyset$.

Formally, given a W-1-q-mUFLP instance

$$< G = (V, A, C, W), s, D, N, \{T_k\}_{k=1,\ldots,N} >,$$

the objective is to find for each $d_r \in D$ a path $P_{d_r} = \{i_1 = s, \ldots, i_N = d_r\} \in T_1 \times \cdots \times T_N$ from $s \in T_1$ to d_r corresponding to the minimum total cost, i.e., to choose $\emptyset \neq S_k \subset T_k, k = 1, \ldots, N$ such that

$$\sum_{d_r \in D} \min_{P_r \in S_1 \times \cdots \times S_l} \Lambda(P_r) + \sum_{k=1}^{N} \sum_{i_k \in S_k} w_{i_k}.$$

is minimized.

Theorem 6 claims that the W-1-q-mUFLP is polynomially Karp-reducible to the classical (metric) multilevel uncapacitated facility location problem (l-UFLP).

Theorem 6. *W-1-q-mUFLP* $<_m^p$ *l-UFLP.*

Proof. To prove the thesis, a polynomial-time reduction algorithm must be found that transforms any W-1-q-mUFLP instance into a l-UFLP instance and vice versa.

Once assimilated the setting up facility costs $f(\cdot)$ with the node weight function $W(\cdot)$ and the set of clients D with the set of destinations D, the polynomial reduction can be proved following similar reasonings as for the claim of Theorems 4 and 5.

The proof is completed by observing that the l-UFLP instance $< D \cup \mathcal{F}, C, f(\cdot) >$ admits a solution made of $|D|$ open paths $P_j = \{i_1, \ldots, i_l\} \in \mathcal{P} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_l$ $(j = 1, \ldots, |D|)$ of l facilities with exactly one from each of the l levels and having connection cost $\sum_{j \in D} \Gamma_{jP}$ and total cost

$$\sum_{j \in D} \Gamma_{jP} + \sum_{k=1}^{l} \sum_{i_k \in S_k} f_{i_k}$$

if and only if the W-1-q-mUFLP instance $< G = (V, A, C, W), s, D, N, \{T_k\}_{k=1,\ldots,N} >$ admits a solution made of $|D|$ paths P_r from s to $d_r \in D$ $(r = 1, \ldots, |D|)$. Each path P_r has $l(P_r) = \Lambda(P_r) = \Gamma_{d_r P}$ and the total cost of the solution is given by

$$\sum_{d_r \in D} \Gamma_{d_r P} + \sum_{k=1}^{N} \sum_{i_k \in S_k} w_{i_k}. \qquad \square$$

A consequence of Theorem 6 is that W-1-q-mUFLP can be formulated as the following linear programming problem.

Let be $\mathcal{T} = \cup_{k=1}^{N} T_k$ and let us define a Boolean decision vector $y \in \{0, 1\}^{|\mathcal{T}|}$ s.t. $\forall\, i_k \in \mathcal{T}_k, k = 1, \ldots, N$

$$y_{i_k} = \begin{cases} 1, \text{ if node } i_k \text{ is in } P_r \text{ for some } r \in \{1, \ldots, |D|\}; \\ 0, \text{ otherwise.} \end{cases}$$

Introducing a Boolean decision variable $x_{d_r P}$ for each path $P \in \mathcal{P}$ and each destination node $d_r \in D$ s.t.

$$x_{d_r P} = \begin{cases} 1, \text{ if node } d_r \text{ is the terminal node of } P_r; \\ 0, \text{ otherwise,} \end{cases}$$

then,

$$\text{(W-1-}q\text{-mUFLP)} \min \sum_{P \in \mathcal{P}} \sum_{d_r \in D} \Gamma_{d_r P}\, x_{d_r P} + \sum_{k=1}^{N} w_{i_k}\, y_{i_k}$$

$$\text{s.t.}$$

$$\sum_{P \in \mathcal{P}} x_{d_r P} = 1, \quad \forall\, d_r \in D, \tag{7}$$

$$\sum_{P \in \mathcal{P}:\, i_k \in P} x_{d_r P} - y_{i_k} \leq 0, \quad \forall\, d_r \in D, \quad \forall\, i_k \in T_k, \quad k = 1, \ldots, N, \tag{8}$$

$$x_{d_r P} \in \{0, 1\}, \quad \forall\, P \in \mathcal{P}, \quad \forall\, d_r \in D, \tag{9}$$

$$y_{i_k} \in \{0, 1\}, \quad \forall\, i_k \in T_k, \quad k = 1, \ldots, N. \tag{10}$$

Constraint (7) imposes that each destination node d_r is terminal node of at least one path. Constraints (8) guarantee that each destination d_r cannot be terminal node of a path P unless P passes through node i_k. In fact, if $y_{i_k} = 0$, then the sum of all assignment variables for node d_r to use paths containing i_k must also be 0.

Note that the combination of constraints (7) and (9) allows to relax constraints (9) that can be replaced by $x_{d_r P} \geq 0$, $\forall\, P \in \mathcal{P}$, $\forall\, d_r \in D$. Similarly, constraints (10) can be replaced by $y_{i_k} \geq 0$, $\forall\, i_k \in T_k$, $k = 1, \ldots, N$, since the sum in constraints (8) is bounded from above by the sum in constraints (7) and $w_i > 0$, for each $i \in V$.

4 Conclusions and Future Directions

This paper studies the SPTPs, special network flow problems recently proposed in the literature that have originated from applications in combinatorial optimization problems with precedence constraints to be satisfied. In [14], the classical and simplest version of the problem has been proved to be polynomially solvable since it reduces to a special single source–single destination SPP. In that paper, several alternative exact algorithms have been proposed and the results of an extensive computational experience are reported to demonstrate empirically which algorithms result more efficient in finding an optimal solution to several different problem instances. Looking at the results, Dijkstra's algorithm outperforms all the competitors. Nevertheless, further experiments would be needed on a wider set of instances and further investigation is planned in the next future in order to implement and test a collection of different algorithms that

- Use path length upper bounds [4]
- Mix Auction and *graph collapsing* [5] and *virtual sources* [6] ideas
- Use the structure of the SPTP and/or the structure of the expanded graph

In this paper, several different variants of the classical SPTP have been stated and formally described as special facility location problems. This relationship between

the two families of problems suggests that the SPTPs could be a powerful tool usable to attack the location problems. It appears that much work could be done along this direction, both with regard to approximation and exact algorithms for location problems.

In addition, thinking to future research it would be also interesting to study some further variants of the SPTP and their complexity where the constraint $\bigcap_{k=1}^{N} T_k = \emptyset$ is relaxed and/or arc capacity constraints are added.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms and Applications. Prentice-Hall Englewood Cliffs (1993)
2. Bertsekas, D.P.: An auction algorithm for shortest paths. SIAM J. Optim. **1**, 425–447 (1991)
3. Bertsekas, D.P.: Dynamic Programming and Optimal Control. 3rd Edition, Vol. I. Athena Scientific (2005)
4. Bertsekas, D.P., Pallottino, S., Scutellà, M.G.: Polynomial auction algorithms for shortest paths. Comput. Optim. Appl. **4**, 99–125 (1995)
5. Cerulli, R., Festa, P., Raiconi, G.: Graph collapsing in shortest path auction algorithms. Comput. Optim. Appl. **18**, 199–220 (2001)
6. Cerulli, R., Festa, P., Raiconi, G.: Shortest path auction algorithm without contractions using virtual source concept. Comput. Optim. Appl. **26**(2), 191–208 (2003)
7. Cherkassky, B.V., Goldberg, A.V.: Negative-cycle detection algorithms. Math. Prog. **85**, 277–311 (1999)
8. Cherkassky, B.V., Goldberg, A.V., Radzik, T.: Shortest path algorithms: theory and experimental evaluation. Math. Prog. **73**, 129–174 (1996)
9. Cherkassky, B.V., Goldberg, A.V., Silverstein, C.: Buckets, heaps, lists, and monotone priority queues. SIAM J. Comput. **28**, 1326–1346 (1999)
10. Denardo, E.V., Fox B.L.: Shortest route methods: 2. group knapsacks, expanded networks, and branch-and-bound. Oper. Res. **27**, 548–566 (1979)
11. Denardo, E.V., Fox, B.L.: Shortest route methods: reaching pruning, and buckets. Oper. Res. **27**, 161–186 (1979)
12. Deo, N., Pang, C.: Shortest path algorithms: taxonomy and annotation. Networks **14**, 275–323 (1984)
13. Dijkstra E.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
14. Festa, P.: Complexity analysis and optimization of the shortest path tour problem. Optim. Lett.(to appear) 1–13 (2011). doi: 10.1007/s11590-010-0258-y
15. Gallo, G., Pallottino, S.: Shortest path methods: a unified approach. Math. Prog. Study. **26**, 38–64 (1986)
16. Gallo, G., Pallottino, S.: Shortest path methods. Ann. Oper. Res. **7**, 3–79 (1988)
17. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations. Plenum Press (1972)
18. Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization: Algorithms and complexity. Prentice-Hall (1982)
19. Shier, D.R., Witzgall, C.: Properties of labeling methods for determining shortest path trees. J. Res. Nat. Bur. Stand. **86**, 317–330 (1981)

Part III
Game Theory and Cooperative Control Foundations for Dynamics of Information Systems

A Hierarchical MultiModal Hybrid Stackelberg–Nash GA for a Leader with Multiple Followers Game

Egidio D'Amato, Elia Daniele, Lina Mallozzi, Giovanni Petrone, and Simone Tancredi

Abstract In this paper a numerical procedure based on a genetic algorithm (GA) evolution process is given to compute a Stackelberg solution for a hierarchical n+1-person game. There is a leader player who enounces a decision before the others, and the rest of players (followers) take into account this decision and solve a Nash equilibrium problem. So there is a two-level game between the leader and the followers, called Stackelberg–Nash problem. The idea of the Stackelberg-GA is to bring together genetic algorithms and Stackelberg strategy in order to process a genetic algorithm to build the Stackelberg strategy. In the lower level, the followers make their decisions simultaneously at each step of the evolutionary process, playing a so called Nash game between themselves. The use of a multimodal genetic algorithm allows to find multiple Stackelberg strategies at the upper level. In this model the uniqueness of the Nash equilibrium at the lower-level problem has been supposed. The algorithm convergence is illustrated by means of several test cases.

E. D'Amato
Dipartimento di Scienze Applicate, Università degli Studi di Napoli "Parthenope", Centro Direzionale di Napoli, Isola C 4 - 80143 Napoli, Italy
e-mail: egidio.damato@uniparthenope.it

E. Daniele • S. Tancredi
Dipartimento di Ingegneria Aerospaziale, Università degli Studi di Napoli "Federico II", Via Claudio 21 - 80125 Napoli, Italy
e-mail: elia.daniele@unina.it; simone.tancredi@unina.it

G. Petrone
Mechanical Engineering and Institute for Computational Mathematical Engineering Building 500, Stanford University. Stanford, CA 94305-3035
e-mail: gpetrone@stanford.edu

L. Mallozzi (✉)
Dipartimento di Matematica e Applicazioni, Università degli Studi di Napoli "Federico II", Via Claudio 21 - 80125 Napoli, Italy
e-mail: mallozzi@unina.it

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_14,

Keywords Genetic algorithm • Hierarchical game • Nash equilibrium • Stackelberg strategy

1 Introduction

The idea to use genetic algorithms to compute solutions to problems arising in Game Theory can be found in different papers [7, 13, 15, 16]. More precisely, in [15] the authors solve a Stackelberg problem with one leader and one follower (leader–follower model) by using genetic algorithm; in [13] the authors solve with GA a Nash equilibrium problem, the well known solution concept in Game Theory for a n players noncooperative game. Both types of solutions are considered in a special aerodynamics problem by [16].

A more general case, dealing with one leader and multiple followers, is the so-called Stackelberg–Nash problem, largely used in different applicative contexts as Transportation or Oligopoly Theory. The paper by [7] designs a genetic algorithm for solving Stackelberg–Nash equilibrium of nonlinear multilevel programming with multiple followers.

In this paper, we deal with a general Stackelberg–Nash problem and assume the uniqueness of the Nash equilibrium solution of the follower players. We present a genetic algorithm suitable to handle multiple solutions for the leader by using multimodal optimization tools.

In the first stage one of the players, called the leader, chooses an optimal strategy knowing that, at the second stage, the other players react by playing a noncooperative game which admits one Nash equilibrium, while a multiple Stackelberg solution may be managed at upper level. In the same spirit of [15], the followers' best reply is computed at each step. For any individual of the leader's population, in our case, multiple followers compute a Nash equilibrium solution, by using a genetic algorithm based on the classical adjustment process [5]. Then, the best reply Nash equilibrium—supposed to be unique—is given to the leader and an optimization problem is solved. We consider also the possibility that the leader may have more than one optimal solution, so that the multimodal approach based on the sharing function let us to reach all this possible solutions in the hierarchical process.

A step by step procedure for optimization based on genetic algorithms (GA) has been implemented starting from a simple Nash equilibrium, through a Stackelberg solution, up to a hierarchical Stackelberg–Nash game, validated by different test cases, even in comparison with other researchers proposals [15, 16]. A GA is presented for a Nash equilibrium problem in Sect. 2 and for a Stackelberg–Nash problem in Sect. 3, together with test cases. Then some applications of the real life are indicated in the concluding Sect. 4.

1.1 The Stackelberg–Nash Model

Let us consider an n+1 player game, where one player P_0 is the leader and the rest of them $P_1, \ldots, P_n$ are followers in a two-level Stackelberg game. Let $X, Y_1, \ldots, Y_n$ be compact, nonempty, and convex subsets of an Euclidean space that are the leader's and the followers' strategy sets, respectively. Let $l, f_1, \ldots, f_n$ be real-valued functions defined on $X \times Y_1 \times \cdots \times Y_n$ representing the leader's and the followers' cost functions. We also assume that $l, f_1, \ldots, f_n$ are continuous in $(x, y_1, \ldots, y_n) \in X \times Y_1 \times \cdots \times Y_n$ and that f_i is strictly convex in y_i for any $i = 1, \ldots, n$. We assume that players are cost minimizers.

The leader is assumed to announce his strategy $x \in X$ in advance and commit himself to it. For a given $x \in X$ the followers select $(y_1, \ldots, y_n) \in R(x)$ where $R(x)$ is the set of the Nash equilibria of the n-person game with players $P_1, \ldots, P_n$, strategy sets $Y_1, \ldots, Y_n$ and cost functions $f_1, \ldots, f_n$. In the Nash equilibrium solution concept, it is assumed that each player knows the equilibrium strategies of the other players and no player has anything to gain by changing only his own strategy unilaterally [1]. For each $x \in X$, which is the leader's decision, the followers solve the following *lower-level Nash equilibrium problem* $\mathcal{N}(x)$:

$$\begin{cases} \text{find } (\bar{y}_1, \ldots, \bar{y}_n) \in Y_1 \times \cdots \times Y_n \text{ such that} \\ f_1(x, \bar{y}_1, \ldots, \bar{y}_n) = \inf\limits_{y_1 \in Y_1} f_1(x, y_1, \bar{y}_2, \ldots, \bar{y}_n) \\ \ldots \\ f_n(x, \bar{y}_1, \ldots, \bar{y}_n) = \inf\limits_{y_n \in Y_n} f_n(x, \bar{y}_1, \ldots, \bar{y}_{n-1}, y_n). \end{cases}$$

The nonempty set $R(x)$ of the solutions to the problem $\mathcal{N}(x)$ is called the followers' reaction set. The leader takes into account the followers Nash equilibrium, which we assume to be unique, and solves an optimization problem in a backward induction scheme.

Let $(\tilde{y}_1(x), \ldots, \tilde{y}_n(x)) \in Y_1 \times \cdots \times Y_n$ be the unique solution of the problem $\mathcal{N}(x)$, the map

$$x \in X \to R(x) = \{\tilde{y}_1(x), \ldots, \tilde{y}_n(x)\}$$

is called the followers' best reply (or response). The leader has to compute a solution of the following *upper level problem* $\mathcal{S}$:

$$\begin{cases} \text{find } \bar{x} \in X \text{ such that} \\ l(\bar{x}, \tilde{y}_1(\bar{x}), \ldots, \tilde{y}_n(\bar{x})) = \inf\limits_{x \in X} l(x, \tilde{y}_1(x), \ldots, \tilde{y}_n(x)). \end{cases}$$

Any solution $\bar{x} \in X$ to the problem $\mathcal{S}$ is called a Stackelberg–Nash strategy, while any vector $(\bar{x}, \tilde{y}_1(\bar{x}), \ldots, \tilde{y}_n(\bar{x})) \in X \times Y_1 \times \cdots \times Y_n$ is called a Stackelberg–Nash equilibrium.

The given definition for $n = 1$ is nothing but the classical Stackelberg solution [1]. This model, for $n > 1$, has been intensively studied and used in different application contexts, as in [2, 9–11, 14].

The Stackelberg problem is known to have the *reversed* information structure since although the leader announces x first, he acts after the follower [6]. The same can be remarked for the Stackelberg–Nash problem.

Let us point out that the problem $\mathcal{S}$ may have more than one solution: this is the so-called multimodal case. To multiple leader's Stackelberg–Nash strategies correspond different Stackelberg–Nash equilibria. The objective of this paper is to compute all the possible leader's Stackelberg–Nash strategies and then, since we suppose that the followers' reaction set is a singleton, all the possible Stackelberg–Nash equilibria of the game.

1.2 Surviving of the Fittest and GA

Genetic algorithms are adaptive heuristic search algorithms based on darwinian natural evolution processes. In analogy to living organisms in nature, individuals of a population can be managed by computers as a digital—binary or floating point—DNA with the diversity associated to design variables optimization problem [4].

A genetic algorithm consists of a finite population of individuals of assigned size, each of them usually encoded as a string of bits named genotype, an adaptive function, called fitness, which provides a measure of the individual to adapt to the environment, which is an estimate of the goodness of the solution and an indication on the individuals most likely to reproduce, semi-random genetic operators such as selection, crossover, and mutation that operate on the genotype expression of individuals, changing their associated fitness.

Because of the large dimension of global optimization problems and access to low-cost distributed parallel environments—such as clusters of PCs—it would be very useful replacing a global optimization by sharing it in local sub-optimizations by using Game Theory tools.

1.3 Multimodal Optimization

In this paper a common procedure of GAs has been introduced in the algorithm to compute a Stackelberg–Nash strategy in line with [7]. Our approach considers also the case of multiple Stackelberg–Nash strategies for the leader, while for the follower's choice the hypothesis of uniqueness still holds [7, 10, 14].

The multimodal optimization is a technique used when the research of all relative maxima or minima of an objective function is needed.

For such analysis a GA is easily implementable, since it is possible to introduce a so-called "sharing function," i.e., a function that provides a penalty. This penalty is correlated to the distance between an individual in the chromosome and the best

individual in the chromosome, relative to the present generation of the evolution process: in this way the algorithm would be forced to modify its pattern research to explore *"areas"* where could be found other maxima (if present), which is the main target of multimodal optimization.

For each individual of the population we compute a relative distance based, e.g., on chromosome s for a simple case of two design variables

$$d_{i,j} = \sqrt{\sum_{k=1}^{r}(s_{i,k} - s_{j,k})^2}, \quad \forall i, j = 1, \ldots, n$$

where r is the number of design variables (or genes, at least one for each player as written for s above), n the population size, and $d_{i,j}$ the distance between individuals i and j. The *"sharing function"* is defined as follows

$$p_{i,j} = 1 + \log\left(\frac{d_{i,j}}{d_{\min}}\right), \begin{cases} p_{i,j} \geq 1 \Rightarrow p = 1 \\ p_{i,j} \leq 0.01 \Rightarrow p = 0.01 \end{cases}$$

where $d_{\min}$ is the minimum distance allowed between different individuals in the same population and it is based usually on domain extent. At this point a multimodal fitness function for each individual i in the population can be evaluated from the previous fitness function

$$\text{fitness}_i^{mm} = \text{fitness}_i \cdot \prod_{j=1}^{n} p_{i,j}.$$

We will use this function to approach the multimodal case and compute multiple Stackelberg–Nash strategies.

2 Nash Genetic Algorithm

2.1 Algorithm Description

We present here the algorithm for a two-player Nash equilibrium game. Let Y_1, Y_2 be compact subsets denoted by players' strategy sets. Let f_1, f_2 be two real-valued functions defined on $Y_1 \times Y_2$ representing the players' cost functions. The algorithm is based on the Nash *adjustment process* [5], where players take turns setting their outputs, and each player's chosen output is a best response to the output his opponent chose the period before. If the process does converge, the steady state is a Nash equilibrium of the game. A Nash equilibrium problem is solved by the two players.

Let $\mathbf{s} = u, v$ be the string (or individual, or chromosome) representing the potential solution for a two person Nash problem.

Then u denotes the subset of variables handled by player 1, belonging to a metric space Y_1, and optimized under an objective function always denoted by f_1. Similarly v indicates the subset of variables handled by player 2, belonging to metric space Y_2, and optimized along another objective function denoted by f_2. Thus, as advocated by Nash equilibrium definition [12], player 1 optimizes the chromosome with respect to the first objective function by modifying u while v is fixed by player 2; symmetrically, player 2 optimizes the chromosome with respect to the second criterion by modifying v , while u is fixed by player 1.

Let u^{k-1} and v^{k-1} be the best values found by players 1 and 2, respectively, at generation $k-1$. At generation k, player 1 optimizes u^k using v^{k-1} in order to evaluate the chromosome (now $\mathbf{s} = u^k, v^{k-1}$). At the same time player 2 optimizes v^k using u^{k-1} to evaluate his chromosome (in this case $\mathbf{s} = u^{k-l}, v^k$).

The algorithm is organized in several steps that consist of:

1. Creating two different random populations, one for each player only at the first generation. Player 1's optimization task is performed by population 1 and vice versa.
2. The classification is made on the basis of the evaluation of a fitness function, typical of GAs, that accounts for the results of matches between each individual of population 1 with all individuals of population 2, scoring 1 or -1, respectively, for a win or loss, and 0 for a draw.

$$\begin{cases} \text{if } f_1(u_i^k, v^{k-1}) > f_1(u^{k-1}, v_i^k), \text{fitness}_1 = 1 \\ \text{if } f_1(u_i^k, v^{k-1}) < f_1(u^{k-1}, v_i^k), \text{fitness}_1 = -1. \\ \text{if } f_1(u_i^k, v^{k-1}) = f_1(u^{k-1}, v_i^k), \text{fitness}_1 = 0 \end{cases}$$

   Similarly, for player 2:

$$\begin{cases} \text{if } f_2(u_i^k, v^{k-1}) < f_2(u^{k-1}, v_i^k), \text{fitness}_2 = 1 \\ \text{if } f_2(u_i^k, v^{k-1}) > f_2(u^{k-1}, v_i^k), \text{fitness}_2 = -1. \\ \text{if } f_2(u_i^k, v^{k-1}) = f_2(u^{k-1}, v_i^k), \text{fitness}_2 = 0 \end{cases}$$

   In this way a simple sorting criterion could be established. For equal fitness value individual are sorted on objective function f_1 for population 1 (player 1) and on objective function f_2 for player 2.
3. A mating pool for parent chromosome is generated and common GA techniques as crossover and mutation are performed on each player population. A second sorting procedure is needed after this evolution process.
4. At the end of kth generation optimization procedure player 1 communicates his own best value u^k to player 2 who will use it at generation $k+1$ to generate its entire chromosome with a unique value for its first part, i.e., the one depending on player 1, while on the second part comes from common GAs crossover and mutation procedure. Conversely, player 2 communicates its own best value v^k to player 1 who will use it at generation $k+1$, generating a population with a unique value for the second part of chromosome, i.e., the one depending on player 2;

Table 1 GA details

Parameter	Value
Population size [-]	50
Crossover fraction [-]	0.90
Mutation fraction [-]	0.10
Parent sorting	Tournament between couples
Mating pool [%]	50
Elitism	No
Crossover mode	Simulated Binary Crossover (SBX)
Mutation mode	Polynomial
$d_{\min}$ for multimodal [-]	0.2

5. A Nash equilibrium is found when a terminal period limit is reached, after repeating the steps 2–4.

This kind of structure for the algorithm is similar to those used by other researchers, with a major emphasis on fitness function consistency [16].

2.2 Test Case

In this test case and also in all the subsequent ones the characteristics the of GAs algorithm are summarized in Table 1.

For the algorithm validation we consider the following example presented in [16]: the strategy sets are $Y_1 = Y_2 = [-5, 5]$ and

$$f_1(y_1, y_2) = (y_1 - 1)^2 + (y_1 - y_2)^2$$
$$f_2(y_1, y_2) = (y_2 - 3)^2 + (y_1 - y_2)^2$$

for which the analytical solution is

$$y_1{}^N = \frac{5}{3}, y_2{}^N = \frac{7}{3}.$$

By using the proposed algorithm our numerical results are

$$\hat{y}_1^N = 1.6665, \hat{y}_2^N = 2.3332.$$

3 Hierarchical Stackelberg–Nash Genetic Algorithm

3.1 Stackelberg Genetic Algorithm

Here the algorithm is presented for a two-player leader–follower game or Stackelberg game. Let X, Y be compact subsets of metric spaces that are the players' strategy sets. Let l, f be two real-valued functions defined on $X \times Y$ representing the players' cost functions [15]. For any $x \in X$ leader's strategy, the follower solves the problem

$$\begin{cases} \text{find } \bar{y} \in Y \text{ such that} \\ f(x, \bar{y}) = \inf\limits_{y \in Y} f(x, y) \end{cases}$$

If there is a unique solution to this problem, say $\tilde{y}(x)$ for any $x \in X$, the leader's problem is

$$\begin{cases} \text{find } \bar{x} \in X \text{ such that} \\ l(\bar{x}, \tilde{y}(\bar{x})) = \inf\limits_{x \in X} l(x, \tilde{y}(x)). \end{cases}$$

Any $\bar{x} \in X$ solution of the leader's problem is called a Stackelberg strategy; any pair $(\bar{x}, \tilde{y}(\bar{x})) \in X \times Y$ is called Stackelberg equilibrium.

The initial population is provided with a random seeding in the subset of a metric space, X, i.e., that of the leader. For each individual (or chromosome) of the leader population, say u_i, a random population for the follower player is generated, i.e., providing v_i. At this step a typical best reply search for the follower player is made until a terminal period is reached or an exit criterion in case of same fitness function value for the first two "best" individuals in population (i.e., follower player). The result of this first step is to determine the follower player's best reply, which is ready to be passed to the leader player in order to evaluate his own best strategy. The leader population has to be sorted under objective function criterium and a mating pool is generated. Now a second step begins and a common crossover and mutation operation on the leader population is performed. Again the follower's best reply should be computed, in the same way described above (reminding the previous syntax for a chromosome we could say that in this case the individual is made of a string $s_i = u_i, v_i$) since the leader population has changed under evolution process. This is the kernel procedure of the genetic algorithm that is repeated until a terminal period is reached or an exit criterion is met.

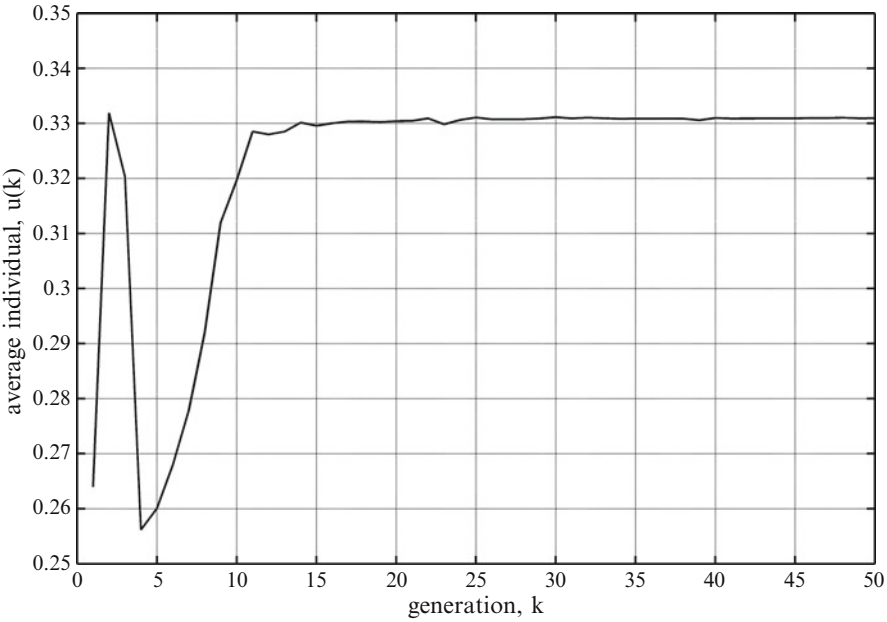

Fig. 1 Average leader's variable history

3.2 Test Case

For the algorithm validation a test case is considered: the strategy sets are $X = Y = [0, 1]$ and

$$l(x, y) = x^2 - y^2 - y$$
$$f(x, y) = xy + (y - x)^2$$

for which $\tilde{y}(x) = \frac{x}{2}$ and the analytical solution is

$$x^S = \frac{1}{3}, y^S = \frac{1}{6}$$

while the numerical solution is

$$\hat{x}^S = 0.3315,\ \hat{y}^S = 0.1672.$$

See Fig. 1 for the average leader's variable history and Fig. 2 for the best leader's variable history.

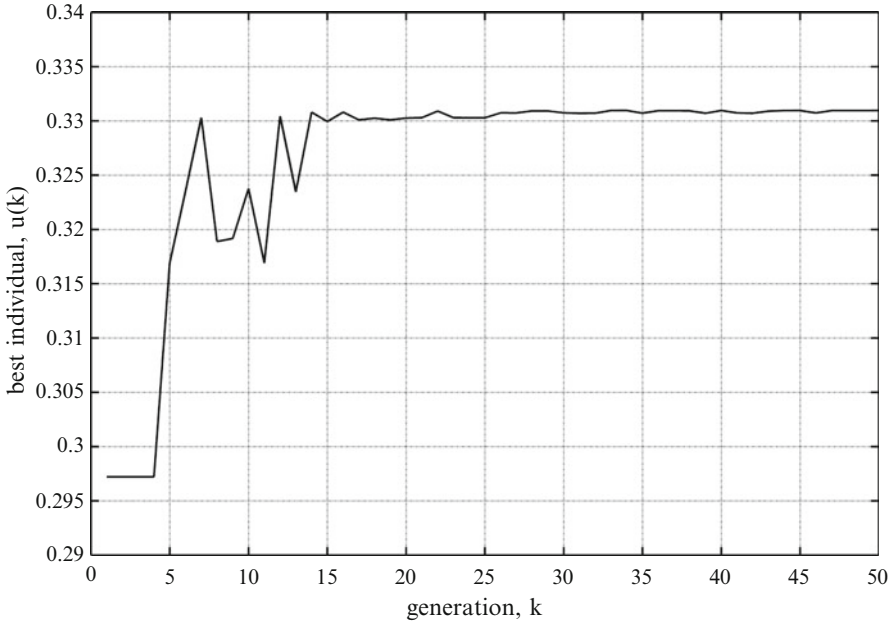

Fig. 2 Best leader's variable history

3.3 Hierarchical Algorithm Description

The initial population for the players is provided with a random seeding in X, the leader's strategy space. For each individual (or chromosome) of the leader population, say u_i, a random population for each of the follower players is generated, i.e., providing $v_{1,i}, \ldots, v_{n,i}$. The only difference from the simple Stackelberg one-leader/one-follower is that now a typical Nash game is played between follower players until they reach a terminal period or an exit criterium. An approach similar to that described before for simple Stackelberg is used also in this case, by considering the algorithm described in Sect. 2.1 for the followers problem and the one described in Sect. 3.1 for the two-level problem (see Fig. 3 for the algorithm structure).

Under our assumption on the cost functions and strategy spaces, we have that the set of Nash equilibria of the lower-level problem $\mathcal{N}(x)$ is nonempty for any $x \in X$. Since we assume also the uniqueness of the equilibrium it turns out that the best reply map $x \in X \to (\tilde{y}_1(x), \ldots, \tilde{y}_n(x))$ is a continuous function. The followers will use at every step this best reply and the genetic algorithm will end up to a Stackelberg–Nash solution, which is any one given by

$$\text{argmin}_{x \in X} l(x, \tilde{y}_1(x), \ldots, \tilde{y}_n(x))$$

not necessarily unique.

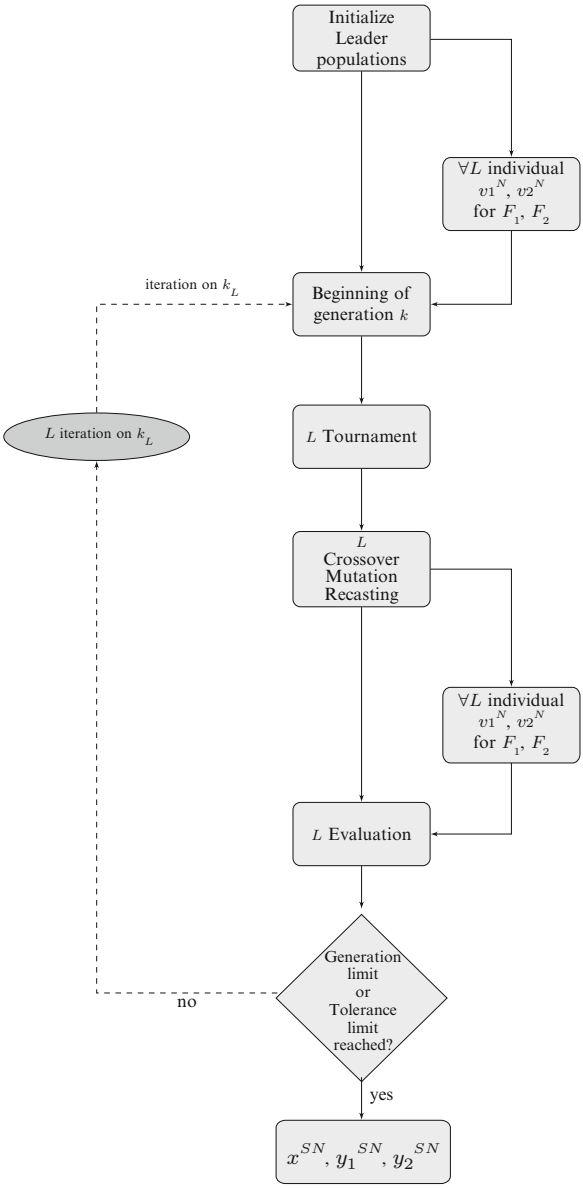

Fig. 3 Stackelberg–Nash algorithm structure

3.4 Test Case

For the algorithm validation we consider a test case with a one leader two followers Stackelberg game (n=2): the strategy sets are $X = Y_1 = Y_2 = [0, 1]$ and

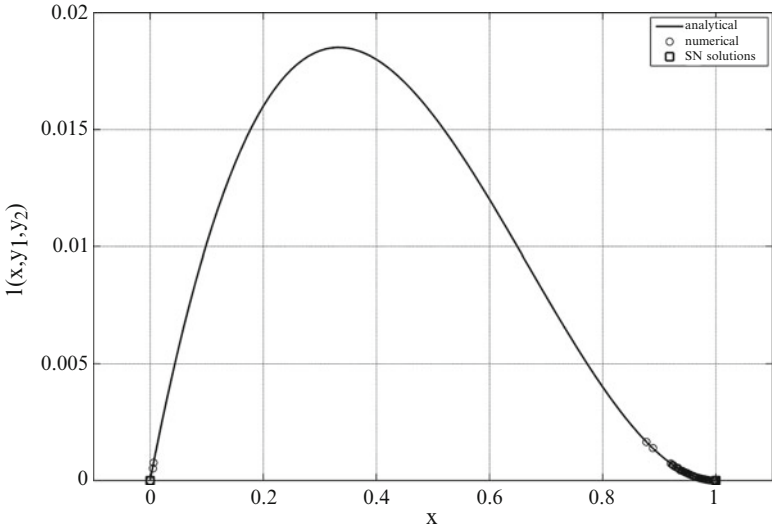

Fig. 4 Leader's cost function history

$$l(x, y_1, y_2) = y_2 \left(x - y_1 - \frac{1}{2} \right)^2$$

$$f_1(x, y_1, y_2) = (y_2 - y_1)^2 + (2y_1 - x)^2$$

$$f_2(x, y_1, y_2) = (y_1 - y_2)^2 + \frac{(2y_2 - x)^2}{2}$$

for which $(\tilde{y}_1(x), \tilde{y}_2(x)) = (x/2, x/2)$ is a Nash equilibrium of the lower-level problem for each $x \in X$. The upper level problem

$$\begin{cases} \text{find } \bar{x} \in X \text{ such that} \\ l(\bar{x}, \tilde{y}_1(\bar{x}), \tilde{y}_2(\bar{x})) = \inf\limits_{x \in X} l(x, \tilde{y}_1(x), \tilde{y}_2(x)) = \inf\limits_{x \in X} \dfrac{x}{8}(x-1)^2 \end{cases}$$

has two solutions $\bar{x} = 0$ and $\bar{x} = 1$. The analytical Stackelberg–Nash solutions are

$$x^{SN} = 1, y_1^{SN} = \frac{1}{2}, y_2^{SN} = \frac{1}{2}$$

$$x_*^{SN} = 0, y_{1*}^{SN} = 0, y_{2*}^{SN} = 0$$

while the numerical procedure has led us to

$$\hat{x}^{SN} = 1.000, \hat{y}_1^{SN} = 0.4999, \hat{y}_2^{SN} = 0.4997$$

$$\hat{x}_*^{SN} = 0, \hat{y}_{1*}^{SN} = 0, \hat{y}_{2*}^{SN} = 0.$$

The leader's cost function history is shown in Fig. 4.

4 Final Remarks

4.1 Applications to Real-Life Problems

The methodology and the computational algorithm developed in this paper could be very useful in many engineering problems, like optimization tool alternative to classical gradient (see also [8, 13, 16]). In particular the hierarchical Stackelberg–Nash model would be very attractive for real engineering problems like the optimization of the position of a multielement airfoil or wing: in these cases important targets for all the movable surfaces that constitute the airfoil (like slats and flaps) or wing (aileron, rudder, elevator, etc.) are present. This kind of methodology (leader–follower) would avoid the traditional troubles faced in multi-objective optimization problem where a scalarization approach is needed to weight every request (objective function) to form a "new" scalar function to optimize. Among others, it would be also very useful in a classic aircraft preliminary design where some specifications appear to be dominant (leader) with respect to others (followers) in determining aircraft shape and dimensions.

Applications in other contexts must be mentioned too: the hierarchical Stackelberg–Nash model has been used, for example, in oligopolies as in [14] or in Transportation problems as in [10].

4.2 Conclusions and Future Works

The genetic algorithm presented in this paper has given very reliable results, also compared with studies of previous literature [16]. An extension to multiple leaders is studied in [3], where it is assumed that the leaders compete in a noncooperative way solving a Nash equilibrium problem.

When a priori the uniqueness of the Nash equilibrium for the lower-level problem cannot be verified, a more robust strategy should be implemented. In this case the upper level problem can be formulated in different ways depending on the leaders' behavior. For example, a security strategy for the leaders could be defined [1]. The multimodal approach proposed in this paper will be useful to build the computational procedure in the case of multiple solutions to the lower-level problem. These aspects, in order to obtain a GA to compute a Stackelberg–Nash equilibrium in a general framework, will be investigated in a future paper.

References

1. Başar, T., Olsder, G.J.: Dynamic noncooperative game theory, Reprint of the second (1995) edition. Classics in Applied Mathematics, 23. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1999)
2. Chinchuluun, A., Pardalos, P.M., Huang, H-X.: Multilevel (Hierarchical) optimization: complexity issues, optimality conditions, algorithms. In: Gao, D., Sherali, H. (eds.) Advances in Applied Mathematics and Global Optimization, pp. 197–221. Springer USA (2009)
3. D'Amato, E., Daniele, E., Mallozzi, L., Petrone, G.: Stackelberg-Nash solutions for global emission games with genetic algorithm, Preprint n. 19 Dipartimento di Matematica e Applicazioni Universita di Napoli di Napoli Federico II (2010)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 181–197 (2002)
5. Fudenberg, D., Tirole, J.: Game Theory. The MIT Press, Cambridge, Massachusetts (1993)
6. Ho, Y.C., Luh, P.B., Muralidharan, R.: Information strucutre, stackelberg games, and incentive, controllability. IEEE Trans. Automat. Control **26**, 454–460 (1981)
7. Liu, B.: Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. Computers Math. Applic. **36**(7), 79–89 (1998)
8. Liu, W. and Chawla, S.: A game theoretical model for adversarial learning. 2009 IEEE International Conference on Data Mining Workshops Miami, Florida, USA, pp. 25–30 (2009)
9. Luo, Z-Q., Pang, J-S., Ralph, D.: Mathematical Programs with Equilibrium Constraints. Cambridge University Press, Cambridge (1996)
10. Marcotte, P., Blain, M.A: Stackelberg-Nash model for the design of deregulated transit system, dynamic games in economic analysis. In: Hamalainen, R.H., Ethamo, H.K. (eds.) Lecture Notes in Control and Information Sciences, vol. 157, pp. 21–28. Springer, Berlin (1991)
11. Migdalas, A., Pardalos, P.M., Varbrand, P. (eds.): Multilevel Optimization: Algorithms and Applications. Kluwer Academic Publishers Kluwer Academic Publishers, Boston USA (1997)
12. Nash, J.: Non-cooperative games. Ann. Math. **54**, 286–295 (1951)
13. Periaux, J., Chen, H.Q., Mantel, B., Sefrioui, M., Sui, H.T.: Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. Finite Elem. Anal. Des. **37**, 417–429 (2001)
14. Sheraly, H.D., Soyster, A.L., Murphy, F.H.: Stackelberg-Nash-Cournot equilibria: characterizations and computations. Operation Res. **31**, 253–276 (1983)
15. Vallée, T., Başar, T.: Off-line computation of Stackelberg solutions with the genetic algorithm. Comput. Econ. **13**, 201–209 (2001)
16. Wang, J.F., Periaux, J.: Multi-Point optimization using GAS and Nash/Stackelberg games for high lift multi-airfoil design in aerodynamics. In: Proceedings of the 2001 Congress on Evolutionary Computation CEC 2001, May 2001, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27–30, pp. 552–559

The Role of Information in Nonzero-Sum Differential Games

Meir Pachter and Khanh Pham

Abstract Games are an ideal vehicle for showcasing the crucial role information plays in decision systems. Indeed, game theory plays a major role in economic theory, and, in particular, in microeconomic theory—one aptly refers to *information economics*. The role of information is further amplified in dynamic games. One then refers to the *information pattern* of the game. In this paper nonzero-sum differential games are addressed and open-loop and state feedback information patterns are considered. Nash equilibria (NE) when complete state information is available and feedback strategies are sought are compared to open-loop NE. In contrast to optimal control and, remarkably, zero-sum differential games, in nonzero-sum differential games the optimal trajectory and the players' values when closed-loop strategies are used are not the same as when open-loop strategies are used. This is amply illustrated in the special case of nonzero-sum Linear-Quadratic differential games. Results which quantify the cost of uncertainty are derived and insight into the dynamics of information systems is obtained.

Keywords Nonzero-sum differential games • Linear-quadratic differential games • Open-loop control • State feedback strategies

M. Pachter (✉)
Air Force Institute of Technology, AFIT, Wright Patterson AFB, OH 45433, USA
e-mail: meir.pachet@afit.edu

K. Pham
Air Force Research Laboratory, Kirtland AFB, Albuquerque, NM 87117, USA
e-mail: khanh.pham@kirtland.af.mil

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_15,

1 Introduction

Nonzero-sum differential games have not received as much attention as zero-sum differential games. In this paper nonzero-sum differential games are addressed. Complete state information is assumed, that is, Nash equilibria (NE) using state feedback strategies are sought. In addition, open-loop strategies and open-loop NE are also discussed. We are interested in gaining insight into the relationship between open-loop and closed-loop NE strategies, and the attendant value functions of the players.

The players are P and E. The dynamics, jointly affected by the respective P and E controls u and v, are

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x, u, v),\ x(0) = x_0,\ 0 \leq t \leq T \tag{1}$$

The state vector $x \in R^n$ and the controls $u \in R^{m_{\mathrm{P}}}$ and $v \in R^{m_{\mathrm{E}}}$.

The P and E players' respective cost functionals are

$$J^{(\mathrm{P})}(u, v; x_0) = G^{(\mathrm{P})}(x(T)) + \int_0^T L^{(\mathrm{P})}(t, x, u, v)\mathrm{d}t \tag{2}$$

and

$$J^{(\mathrm{E})}(u, v; x_0) = G^{(\mathrm{E})}(x(T)) + \int_0^T L^{(\mathrm{E})}(t, x, u, v)\mathrm{d}t \tag{3}$$

For the sake of time consistency/subgame perfection, a Nash equilibrium (NE) in feedback strategies is normally sought. In our article, both state feedback control strategies $u(t, x)$ and $v(t, x)$, and open-loop strategies $u(t; x_0)$ and $v(t; x_0)$ strategies where only the initial state information x_0 is available to the players, are of interest. The synthesis of optimal state feedback control laws/strategies calls for the application of the method of dynamic programming (DP). The application of DP to deterministic differential games leads to hyperbolic partial differential equations (PDEs) of Hamilton–Jacobi–Bellman (HJB) type. The solution of the HJB PDEs is conducive to the synthesis of optimal state feedback strategies. The solution of the nonzero-sum open-loop differential game entails the solution of two one-sided optimal control problems via the application of the Pontryagin maximum principle (PMM). When open-loop optimal controls are sought, the PMM is evoked, and a TPBVP must be solved.

The first to consider nonzero-sum differential games was James Case; in [1,2], an attempt was made to extend optimal control methods to differential games and open-loop strategies were devised. In the seminal papers [3] and [4] the synthesis of NE using closed-loop controls, namely, state feedback strategies was undertaken. This resulted in a modification of the costate equations where an additional, interaction, or, cross-effect, term was introduced. We refer to (11) in [3] and (12) in [4]. Consequently, it is stated in [3] that concerning the extension of optimal control

methods to nonzero-sum differential games, it would appear that "in the N-player, nonzero-sum game, they are a set of partial differential equations, generally very difficult to solve." Furthermore, it is stated in [4]: "The presence of the summation term in (12) makes the necessary conditions (7), (9), (10), (12), virtually useless for deriving computational algorithms." Attempts were made to apply the theory developed in [3] and [4]—we quote from the application paper [5]: "Recent technological advances in computerized traffic information systems convince us that the feedback Nash equilibrium concept is more realistic and attractive than the open-loop one. However, due to the presence of the cross-effect terms in the adjoint equations, it is extremely difficult to solve the differential game in the feedback strategy space. In fact, a resulting two point boundary value problem is comprised of coupled partial differential equations. Unless the game has a simple structure, its solution is still hard to obtain either analytically or numerically. This is why computational results of open-loop problems have been more frequently reported in the literature despite the fact that feedback solutions are more realistic." Thus, in the follow-up on paper [6], closed-loop strategies were abandoned from the outset.

In [4] a discrete, finite state, two stages, dynamical system is used to show that the open-loop and closed-loop NE and attendant strategies are not the same in a nonzero-sum dynamic game. This point is also taken up in [7] where a nonlinear, but scalar, nonzero-sum advertising differential game is discussed. It is shown that in some regions of the game's state space the feedback representation of the open-loop differential game's NE strategies does not yield the NE strategies in the closed-loop differential game. In this article, this point is illustrated in the context of a LQ nonzero-sum differential game.

The article is organized as follows. In Sect. 2 the HJB PDEs for nonzero-sum differential games are derived and their solution in the special LQ case is given in Sect. 3. Open-loop NE in nonzero-sum differential games are discussed in Sect. 4 and their solution in the special LQ case is given in Sect. 5. Asymmetric information patterns, where, e.g., player P uses feedback strategies and player E uses open-loop controls, are discussed in Sect. 6, followed by concluding remarks in Sect. 7. It is remarkable that in nonzero-sum LQ differential games not only are the open-loop and closed-loop NE and the attendant optimal strategies different, but also, in addition, the values/costs of both players are *lower* when they use open-loop strategies. Furthermore, if just one player uses feedback strategies, the costs of both players end up higher.

2 HJB PDEs and Closed-Loop Strategies

Nonzero-sum differential games where the players have access to complete state information and therefore use closed-loop strategies, are addressed first.

We form the two Hamiltonians

$$H^{(\mathrm{P})}(t,x,u,v,\lambda) = L^{(\mathrm{P})}(t,x,u,v) + \lambda^T \cdot f(t,x,u,v) \tag{4}$$

and

$$H^{(\mathrm{E})}(t,x,u,v,\lambda) = L^{(\mathrm{E})}(t,x,u,v) + \lambda^T \cdot f(t,x,u,v) \tag{5}$$

where the vector $\lambda \in R^n$.

Applying the *principle of optimality* of dynamic programming (DP) for the stated purpose of finding a Nash equilibrium in state feedback strategies, yields the pair of coupled optimization problems

$$-\frac{\partial V^{(\mathrm{P})}}{\partial t} = \min_u H^{(\mathrm{P})}(t,x,u,v,V_x^{(\mathrm{P})}) \ , \quad V^{(\mathrm{P})}(T,x) = G^{(\mathrm{P})}(x) \tag{6}$$

and

$$-\frac{\partial V^{(\mathrm{E})}}{\partial t} = \min_v H^{(\mathrm{E})}(t,x,u,v,V_x^{(\mathrm{E})}) \ , \quad V^{(\mathrm{E})}(T,x) = G^{(\mathrm{E})}(x) \tag{7}$$

where the P and E players' value functions are $V^P(t,x)$ and $V^E(t,x)$, respectively, and subscripts denote partial derivatives. At time $0 \leq t \leq T$ and state $x \in R^n$ the Nash equilibrium of the *static* nonzero-sum game with respective P and E players cost functions $H^{(\mathrm{P})}(t,x,u,v,V_x^{(\mathrm{P})})$ and $H^{(\mathrm{E})}(t,x,u,v,V_x^{(\mathrm{E})})$ is computed:

$$u^* = \arg\min_u H^{(\mathrm{P})}(t,x,u,v^*,V_x^{(\mathrm{P})}) \tag{8}$$

and

$$v^* = \arg\min_v H^{(\mathrm{E})}(t,x,u^*,v,V_x^{(\mathrm{E})}); \tag{9}$$

Hence, we solve the set of two equations in u and v

$$\frac{\partial H^{(\mathrm{P})}(t,x,u,v,V_x^{(\mathrm{P})})}{\partial u} = 0 \tag{10}$$

and

$$\frac{\partial H^{(\mathrm{E})}(t,x,u,v,V_x^{(\mathrm{E})})}{\partial v} = 0 \tag{11}$$

In the static game (8) and (9) the P and E players' respective cost functions $H^{(\mathrm{P})}$ and $H^{(\mathrm{E})}$ are parametrized by $V_x^{(\mathrm{P})}$ *and* $V_x^{(\mathrm{E})}$, respectively. Hence, the solution of the static Nash game yields the optimal "control laws"

$$u^* = \phi(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}) \tag{12}$$

and

$$v^* = \psi(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}); \tag{13}$$

we have used quotation marks to emphasize that the above equations should not be considered control laws/strategies, because the partial derivatives of the value function are not yet known. The respective P and E feedback strategies $u^*(t,x)$ and $v^*(t,x)$ have yet to be synthesized. This point is sometimes obfuscated in the literature and the feedback strategies are prematurely included in the Hamiltonians—see, for example, [8].

Inserting the "control laws" (12) and (13) into the equations of DP (6) and (7) yields a set of two coupled HJB PDEs for the P and E players' value functions $V^{(\mathrm{P})}(t,x)$ and $V^{(\mathrm{E})}(t,x)$:

$$\begin{aligned} -\frac{\partial V^{(\mathrm{P})}}{\partial t} &= H^{(\mathrm{P})}\left(t,x,\phi\left(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}\right),\psi\left(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}\right),V_x^{(\mathrm{P})}\right) \\ V^{(\mathrm{P})}(T,x) &= G^{(\mathrm{P})}(x)\,,\quad 0\le t\le T,\ x\in R^n \end{aligned} \tag{14}$$

and

$$\begin{aligned} -\frac{\partial V^{(\mathrm{E})}}{\partial t} &= H^{(\mathrm{E})}\left(t,x,\phi\left(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}\right),\psi\left(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}\right),V_x^{(\mathrm{E})}\right) \\ V^{(\mathrm{E})}(T,x) &= G^{(\mathrm{E})}(x)\,,\quad 0\le t\le T,\ x\in R^n \end{aligned} \tag{15}$$

The functions ϕ and ψ in the RHS of (14) and (15) are known. In principle, one might obtain the players' value functions and consequently their optimal strategies upon solving the system of two coupled nonlinear PDEs, (14) and (15) by stepping back in time. This is summarized in

Proposition 1. *The solution of the nonzero-sum differential game (1)–(3) entails the solution of two coupled HJB PDEs, (14) and (15), for the P and E players' value functions. Having solved the PDEs (14) and (15), the respective P and E NE state feedback strategies are given by (12) and (13), namely,*

$$u^*(t,x)=\phi\left(t,x,V_x^{(\mathrm{P})}(t,x),V_x^{(\mathrm{E})}(t,x)\right)$$

and

$$v^*(t,x)=\psi\left(t,x,V_x^{(\mathrm{P})}(t,x),V_x^{(\mathrm{E})}(t,x)\right)$$

3 Nonzero-Sum LQ Differential Games

A NE in feedback strategies is sought. We therefore employ the method of DP, which entails the solution of the system of HJB PDEs developed in Sect. 2, (14) and (15). This is demonstrated in the context of LQ differential games.

The dynamics are

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + Bu + Cv,\ x(0) = x_0,\ 0 \le t \le T \tag{16}$$

where A is a $n \times n$ matrix, B is a $n \times m_{\mathrm{P}}$ matrix, and C is a $n \times m_{\mathrm{E}}$ matrix.

The P and E players' cost functionals (2) and (3) are specified as follows:

$$G^{(\mathrm{P})}(x) = x^T Q_F^{(\mathrm{P})} x,\ \ L^{(\mathrm{P})}(t,x,u,v) = x^T Q^{(\mathrm{P})} x + u^T R^{(\mathrm{P})} u \tag{17}$$

and

$$G^{(\mathrm{E})}(x) = -x^T Q_F^{(\mathrm{E})} x,\ \ L^{(\mathrm{E})}(t,x,u,v) = -x^T Q^{(\mathrm{E})} x + v^T R^{(\mathrm{E})} v, \tag{18}$$

respectively. The matrices $R^{(\mathrm{P})}$, $Q^{(\mathrm{P})}$, $Q_F^{(\mathrm{P})}$, $R^{(\mathrm{E})}$, $Q^{(\mathrm{E})}$, and $Q_F^{(\mathrm{E})}$ are real, symmetric, and positive definite. The time dependence of the matrices in (16)–(18) is suppressed.

The specified cost functionals capture the pursuit-evasion character of the game under consideration—whence the P and E designation of the players. At the same time, it is noteworthy that the LQ game (16)–(18) cannot be reduced to a zero-sum game by a proper choice of the parameters/matrices of the cost functionals (17) and (18).

We proceed according to Proposition 1.

The Hamiltonians are

$$H^{(\mathrm{P})}(t,x,u,v,\lambda) = x^T Q^{(\mathrm{P})} x + u^T R^{(\mathrm{P})} u + \lambda^T \cdot (Ax + Bu + Cv) \tag{19}$$

and

$$H^{(\mathrm{E})}(t,x,u,v,\lambda) = -x^T Q^{(\mathrm{E})} x + v^T R^{(\mathrm{E})} v + \lambda^T \cdot (Ax + Bu + Cv) \tag{20}$$

Consequently, the functions

$$\phi\left(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}\right) = -\frac{1}{2}\left(R^{(\mathrm{P})}\right)^{-1} B^T V_x^{(\mathrm{P})}, \tag{21}$$

$$\psi\left(t,x,V_x^{(\mathrm{P})},V_x^{(\mathrm{E})}\right) = -\frac{1}{2}\left(R^{(\mathrm{E})}\right)^{-1} C^T V_x^{(\mathrm{E})}, \tag{22}$$

and the coupled HJB PDEs (14) and (15) in the P and E players' value functions $V^{(\mathrm{P})}(t,x)$ and $V^{(\mathrm{E})}(t,x)$ are

$$\begin{aligned} -\frac{\partial V^{(\mathrm{P})}}{\partial t} &= x^T Q^{(\mathrm{P})} x + x^T A^T V_x^{(\mathrm{P})} - \frac{1}{4}(V_x^{(\mathrm{P})})^T B (R^{(\mathrm{P})})^{-1} B^T V_x^{(\mathrm{P})} \\ &\quad -\frac{1}{2}(V_x^{(\mathrm{E})})^T C (R^{(\mathrm{E})})^{-1} C^T V_x^{(\mathrm{P})}, \\ V^{(\mathrm{P})}(T,x) &= x^T Q_F^{(\mathrm{P})} x,\ T \ge t \ge 0,\ x \in R^n \end{aligned} \tag{23}$$

and

$$
\begin{aligned}
-\frac{\partial V^{(\mathrm{E})}}{\partial t} &= -x^T Q^{(\mathrm{E})} x + x^T A^T V_x^{(\mathrm{E})} - \frac{1}{4}(V_x^{(\mathrm{E})})^T C(R^{(\mathrm{E})})^{-1} C^T V_x^{(\mathrm{E})} \\
&\quad -\frac{1}{2}(V_x^{(\mathrm{P})})^T B(R^{(\mathrm{P})})^{-1} B^T V_x^{(\mathrm{E})}, \\
V^{(\mathrm{E})}(T,x) &= -x^T Q_F^{(\mathrm{E})} x,\ T \geq t \geq 0,\ x \in R^n
\end{aligned} \tag{24}
$$

Claim A
The P and E players' value functions are quadratic in the state x, namely

$$
V^{(\mathrm{P})}(t,x) = x^T P^{(\mathrm{P})}(t) x \tag{25}
$$

and

$$
V^{(\mathrm{E})}(t,x) = x^T P^{(\mathrm{E})}(t) x; \tag{26}
$$

$P^{(\mathrm{P})}(t)$ and $P^{(\mathrm{E})}(t)$ are real, symmetric matrices and $P^{(\mathrm{P})}(t) > 0\ \forall\ 0 \leq t \leq T$.

Inserting (25) and (26) into (23) and (24) yields a set of two coupled matrix Riccati equations

$$
\begin{aligned}
\dot{P}^{(\mathrm{P})} &= A^T P^{(\mathrm{P})} + P^{(\mathrm{P})} A - P^{(\mathrm{P})} B(R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})} + Q^{(\mathrm{P})} \\
&\quad -P^{(\mathrm{P})} C(R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})} - P^{(\mathrm{E})} C(R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{P})}, \\
P^{(\mathrm{P})}(0) &= Q_F^{(\mathrm{P})}, 0 \leq t \leq T
\end{aligned} \tag{27}
$$

and

$$
\begin{aligned}
\dot{P}^{(\mathrm{E})} &= A^T P^{(\mathrm{E})} + P^{(\mathrm{E})} A - P^{(\mathrm{E})} C(R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})} - Q^{(\mathrm{E})} \\
&\quad -P^{(\mathrm{E})} B(R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})} - P^{(\mathrm{P})} B(R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{E})}, \\
P^{(\mathrm{E})}(0) &= -Q_F^{(\mathrm{E})}, 0 \leq t \leq T
\end{aligned} \tag{28}
$$

Upon solving the set of matrix Riccati equations (27) and (28), the respective NE *strategies* of P and E are obtained:

$$
u^*(t,x) = -(R^{(\mathrm{P})}(T-t))^{-1}(B(T-t))^T P^{(\mathrm{P})}(T-t) \cdot x \tag{29}
$$

and

$$
v^*(t,x) = -(R^{(\mathrm{E})}(T-t))^{-1}(C(T-t))^T P^{(\mathrm{E})}(T-t) \cdot x \tag{30}
$$

These results are summarized in

Theorem 1. *A (unique) solution to the closed-loop nonzero-sum LQ differential game (16)–(18) exists* $\forall\ x_0 \in R^n$ *iff a solution on the interval* $0 \leq t \leq T$ *of the two coupled Riccati-type matrix differential equations (27) and (28) exists. The respective NE state feedback strategies of P and E are given by (29) and (30). The respective values of P and E are*

$$V^{(\mathrm{P})}(x_0) = x_0^T P^{(\mathrm{P})}(T)x_0$$

and

$$V^{(\mathrm{E})}(x_0) = x_0^T P^{(\mathrm{E})}(T)x_0$$

Evidently, the solution of the closed-loop nonzero-sum LQ differential game hinges on the solution of the system of Riccati equations (27) and (28) on the interval $0 \leq t \leq T$. A solution always exists for T sufficiently small.

3.1 Minimum Energy Control

The special case where the state error penalty matrices in the cost functionals,

$$Q^{(\mathrm{P})} = 0, \quad Q^{(\mathrm{E})} = 0$$

is of interest; one then refers to minimum energy control. Clearly, the pursuer wants to minimize the P–E separation at the final time T, namely, P wants to minimize the miss distance while at the same time keeping a lid on his expanded control energy. Naturally, the evader's goal is diametrically opposite—he wants to maximize the miss distance without overexerting himself. Note however that even if one were to choose the terminal costs'/miss distance weights $Q_F^{(\mathrm{P})} = Q_F^{(\mathrm{E})}$, this would not be a zero-sum game.

We apply Theorem 2. In the minimum energy pursuit-evasion game the set of coupled Riccati equations (27) and (28) is reduced to the following set of two coupled matrix differential equations

$$\begin{aligned}
\dot{P}^{(\mathrm{P})} &= A^T P^{(\mathrm{P})} + P^{(\mathrm{P})} A - P^{(\mathrm{P})} B (R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})} \\
&\quad - P^{(\mathrm{P})} C (R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})} - P^{(\mathrm{E})} C (R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{P})}, \\
P^{(\mathrm{P})}(0) &= Q_F^{(\mathrm{P})}, 0 \leq t \leq T
\end{aligned} \tag{31}$$

and

$$\begin{aligned}
\dot{P}^{(\mathrm{E})} &= A^T P^{(\mathrm{E})} + P^{(\mathrm{E})} A - P^{(\mathrm{E})} C (R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})} \\
&\quad - P^{(\mathrm{E})} B (R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})} - P^{(\mathrm{P})} B (R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{E})}, \\
P^{(\mathrm{E})}(0) &= -Q_F^{(\mathrm{E})}, 0 \leq t \leq T
\end{aligned} \tag{32}$$

Using the change of variables

$$\Pi^{(\mathrm{P})} \equiv (P^{(\mathrm{P})})^{-1} \tag{33}$$

and

$$\Pi^{(\mathrm{E})} \equiv (P^{(\mathrm{E})})^{-1}, \tag{34}$$

these equations can be transformed into the following form:

$$\begin{aligned}\dot{\Pi}^{(\mathrm{P})} &= -A\Pi^{(\mathrm{P})} - \Pi^{(\mathrm{P})}A^T + B(R^{(\mathrm{P})})^{-1}B^T \\ &\quad + C(R^{(\mathrm{E})})^{-1}C^T(\Pi^{(\mathrm{E})})^{-1}\Pi^{(\mathrm{P})} + \Pi^{(\mathrm{P})}(\Pi^{(\mathrm{E})})^{-1}C(R^{(\mathrm{E})})^{-1}C^T, \\ \Pi^{(\mathrm{P})}(0) &= (Q_F^{(\mathrm{P})})^{-1},\ 0 \le t \le T\end{aligned} \tag{35}$$

and

$$\begin{aligned}\dot{\Pi}^{(\mathrm{E})} &= -A\Pi^{(\mathrm{E})} - \Pi^{(\mathrm{E})}A^T + C(R^{(\mathrm{E})})^{-1}C^T \\ &\quad + B(R^{(\mathrm{P})})^{-1}B^T(\Pi^{(\mathrm{P})})^{-1}\Pi^{(\mathrm{E})} + \Pi^{(\mathrm{E})}(\Pi^{(\mathrm{P})})^{-1}B(R^{(\mathrm{P})})^{-1}B^T, \\ \Pi^{(\mathrm{E})}(0) &= -(Q_F^{(\mathrm{E})})^{-1},\ 0 \le t \le T,\end{aligned} \tag{36}$$

Thus, upon solving the nonlinear system of symmetric matrix differential equations (35) and (36), the P and E players' value functions are obtained

$$V^{(\mathrm{P})}(t,x) = x^T(\Pi^{(\mathrm{P})}(t))^{-1}x$$

and

$$V^{(\mathrm{E})}(t,x) = x^T(\Pi^{(\mathrm{E})}(t))^{-1}x,$$

respectively; the optimal NE feedback strategies are

$$u^*(t,x) = -(R^{(\mathrm{P})}(T-t))^{-1}(B(T-t))^T(\Pi^{(\mathrm{P})}(T-t))^{-1}\cdot x$$

and

$$v^*(t,x) = -(R^{(\mathrm{E})}(T-t))^{-1}(C(T-t))^T(\Pi^{(\mathrm{E})}(T-t))^{-1}\cdot x$$

A further simplification is possible: in the "symmetric" nonzero-sum minimum energy differential game where the P and E players have similar capabilities, that is, the input matrix $B = C$, the control effort weight of P, $R^{(\mathrm{P})} \equiv R$, the control effort weight of E, $R^{(\mathrm{E})} \equiv \alpha R$, $0 < \alpha$, and $Q_F^{(\mathrm{P})} = Q_F^{(\mathrm{P})} \equiv Q_F$, (35) and (36) are

$$\begin{aligned}\dot{\Pi}^{(\mathrm{P})} &= -A\Pi^{(\mathrm{P})} - \Pi^{(\mathrm{P})}A^T + BR^{-1}B^T \\ &\quad + \frac{1}{\alpha}\left[BR^{-1}B^T\left(\Pi^{(\mathrm{E})}\right)^{-1}\Pi^{(\mathrm{P})} + \Pi^{(\mathrm{P})}\left(\Pi^{(\mathrm{E})}\right)^{-1}BR^{-1}B^T\right], \\ \Pi^{(\mathrm{P})}(0) &= Q_F^{-1},\ 0 \le t \le T\end{aligned}$$

and

$$\dot{\Pi}^{(\mathrm{E})} = -A\Pi^{(\mathrm{E})} - \Pi^{(\mathrm{E})}A^T + \frac{1}{\alpha}BR^{-1}B^T + BR^{-1}B^T(\Pi^{(\mathrm{P})})^{-1}\Pi^{(\mathrm{E})} + \Pi^{(\mathrm{E})}(\Pi^{(\mathrm{P})})^{-1}BR^{-1}B^T,$$
$$\Pi^{(\mathrm{E})}(0) = -Q_F^{-1},\ 0 \le t \le T$$

We note however that whereas in the zero-sum minimum energy differential game the transformation $\Pi(t) = P^{-1}(t)$ reduces the Riccati equation to the (linear) Lyapunov equation

$$\dot{\Pi} = -A\Pi - \Pi A^T + B(R^{(\mathrm{P})})^{-1}B^T - C(R^{(\mathrm{P})})^{-1}C^T,\ \Pi(0) = (Q_F)^{-1},\ 0 \le t \le T,$$

this, unfortunately, is not the case in the nonzero-sum differential game, where the transformations (33) and (34) still render a set of two coupled *nonlinear* matrix differential equations, (35) and (36). In the minimum energy nonzero-sum differential game, and also in the simplified "symmetric" minimum energy nonzero-sum differential game, a closed-form solution is not available. This also applies to the case of scalar dynamics, where the system of two first-order nonlinear differential equations parametrized by $a \in R^1, \alpha > 0$ and $\mu \equiv b^2 q_F / r > 0$,

$$\dot{\Pi}^{(\mathrm{P})} = -2a\Pi^{(\mathrm{P})} + 2\frac{\mu}{\alpha}\frac{\Pi^{(\mathrm{P})}}{\Pi^{(\mathrm{E})}} + \mu,\ \Pi^{(\mathrm{P})}(0) = 1,$$
$$\dot{\Pi}^{(\mathrm{E})} = -2a\Pi^{(\mathrm{E})} + 2\mu\frac{\Pi^{(\mathrm{E})}}{\Pi^{(\mathrm{P})}} + \frac{\mu}{\alpha},\ \Pi^{(\mathrm{E})}(0) = -1,\ 0 \le t \le T,$$

must be solved. In general, nonzero-sum differential games are harder to solve than their zero-sum counterpart.

4 Open-Loop Nash Equilibrium in Nonzero-Sum Differential Games

We now address the solution of the nonzero-sum differential game (1)–(3) using open-loop P and E strategies $u(t; x_0)$ and $v(t; x_0)$, respectively: the information available to the P and E players is the initial state information only, x_0. A NE is sought where the NE strategies of players P and E are the respective controls $u^*(t; x_0)$ and $v^*(t; x_0)$, $0 \le t \le T$.

The PMM applies. We form the Hamiltonians

$$H^{(\mathrm{P})}(t, x, u, \lambda^{(\mathrm{P})}) = L^{(\mathrm{P})}(t, x, u, v^*(t; x_0)) + (\lambda^{(\mathrm{P})})^T \cdot f(t, x, u, v^*(t; x_0)) \qquad (37)$$

and

$$H^{(\mathrm{E})}(t,x,v,\lambda^{(\mathrm{E})}) = L^{(\mathrm{E})}(t,x,u^*(t;x_0),v) + (\lambda^{(\mathrm{E})})^T \cdot f(t,x,u^*(t;x_0),v) \quad (38)$$

A necessary condition for the existence of a NE in open-loop strategies entails the existence of nonvanishing costates $\lambda^{(\mathrm{P})}(t)$ and $\lambda^{(\mathrm{E})}(t)$, $0 \le t \le T$, which satisfy the differential equations

$$\frac{\mathrm{d}\lambda^{(\mathrm{P})}}{\mathrm{d}t} = -H_x^{(\mathrm{P})},\ \lambda^{(\mathrm{P})}(T) = Q_F^{(\mathrm{P})}x(T) \quad (39)$$

and

$$\frac{\mathrm{d}\lambda^{(\mathrm{E})}}{\mathrm{d}t} = -H_x^{(\mathrm{E})},\ \lambda^{(\mathrm{E})}(T) = -Q_F^{(\mathrm{E})}x(T) \quad (40)$$

According to the PMM, a static nonzero-sum game with the P and E players' respective costs (37) and (38) is solved $\forall\ 0 \le t \le T$. The optimal control of P is given by the solution of the equation in u,

$$\frac{\partial H^{(\mathrm{P})}(t,x,u,\lambda^{(\mathrm{P})})}{\partial u} = 0, \quad (41)$$

that is,

$$u^*(t) = \tilde{\phi}(t,x,\lambda^{(\mathrm{P})};v^*(t;x_0)) \quad (42)$$

The optimal control of E is given by the solution of the equation in v,

$$\frac{\partial H^{(\mathrm{E})}(t,x,v,\lambda^{(\mathrm{E})})}{\partial v} = 0, \quad (43)$$

that is

$$v^*(t) = \tilde{\psi}(t,x,\lambda^{(\mathrm{E})};u^*(t;x_0)); \quad (44)$$

The functions $\tilde{\phi}$ and $\tilde{\psi}$ are known and, in principle, one can solve the set of two equations in u^* and v^*, (42) and (44). One obtains the "control laws"

$$u^* = \phi(t,x,\lambda^{(\mathrm{P})},\lambda^{(\mathrm{E})}) \quad (45)$$

and

$$v^* = \psi(t,x,\lambda^{(\mathrm{P})},\lambda^{(\mathrm{E})}) \quad (46)$$

In (45) and (46) the functions ϕ and ψ are known.

In the static game the P and E players' cost functions $H^{(\mathrm{P})}$ and $H^{(\mathrm{E})}$ were parametrized by $\lambda^{(\mathrm{P})}$ and $\lambda^{(\mathrm{E})}$, respectively. Hence, the solutions (45) and (46) of

the static Nash game are also parametrized by $\lambda^{(\mathrm{P})}$ and $\lambda^{(\mathrm{E})}$. That is why we have used quotation marks to emphasize that (45) and (46) should not be considered control laws/strategies, because the costates $\lambda^{(\mathrm{P})}$ and $\lambda^{(\mathrm{E})}$ are not yet determined.

The optimal "control laws" (45) and (46) are inserted into (1), yielding the optimal trajectory

$$\frac{\mathrm{d}x^*}{\mathrm{d}t} = f(t, x^*, \phi(t, x^*, \lambda^{(\mathrm{P})}, \lambda^{(\mathrm{E})}), \psi(t, x^*, \lambda^{(\mathrm{P})}, \lambda^{(\mathrm{E})})), \; x(0) = x_0, \; 0 \leq t \leq T \tag{47}$$

The open-loop Nash equilibrium is found upon solving the TPBVP (39), (40), and (47) using (45) and (46).

5 Nonzero-Sum LQ Games: Open-Loop Control

The theory developed in Sect. 4 is now applied to the solution of the open-loop nonzero-sum LQ differential game (16)–(18).

The Hamiltonians are

$$H^{(\mathrm{P})}(t, x, u, \lambda^{(\mathrm{P})}) = x^T Q^{(\mathrm{P})} x + u^T R^{(\mathrm{P})} u + (\lambda^{(\mathrm{P})})^T \cdot (Ax + Bu + Cv^*(t; x_0)) \tag{48}$$

and

$$H^{(\mathrm{E})}(t, x, v, \lambda^{(\mathrm{E})}) = -x^T Q^{(\mathrm{E})} x + v^T R^{(\mathrm{E})} v + (\lambda^{(\mathrm{E})})^T \cdot (Ax + Bu^*(t; x_0) + Cv) \tag{49}$$

Consequently, the functions

$$\phi(t, x, \lambda^{(\mathrm{P})}, \lambda^{(\mathrm{E})}) = -\tfrac{1}{2}(R^{(\mathrm{P})})^{-1} B^T \lambda^{(\mathrm{P})}, \tag{50}$$

$$\psi(t, x, \lambda^{(\mathrm{P})}, \lambda^{(\mathrm{E})}) = -\tfrac{1}{2}(R^{(\mathrm{E})})^{-1} C^T \lambda^{(\mathrm{E})} \tag{51}$$

Using (39), (40), and (47) we obtain the linear TPBVP (52)–(54):

$$\tfrac{\mathrm{d}\lambda^{(\mathrm{P})}}{\mathrm{d}t} = -A^T \lambda^{(\mathrm{P})} - 2Q^{(\mathrm{P})} x^*, \;\; \lambda^{(\mathrm{P})}(T) = Q_F^{(\mathrm{P})} x(T) \tag{52}$$

$$\tfrac{\mathrm{d}\lambda^{(\mathrm{E})}}{\mathrm{d}t} = -A^T \lambda^{(\mathrm{E})} - 2Q^{(\mathrm{E})} x^*, \;\; \lambda^{(\mathrm{E})}(T) = -Q_F^{(\mathrm{E})} x(T) \tag{53}$$

and

$$\frac{\mathrm{d}x^*}{\mathrm{d}t} = Ax^* - \frac{1}{2}\left[B\left(R^{(\mathrm{P})}\right)^{-1}B^T\lambda^{(\mathrm{P})} + C(R^{(\mathrm{E})})^{-1}C^T\lambda^{(\mathrm{E})}\right],\ x(0) = x_0,\ 0 \le t \le T \tag{54}$$

In order to avoid the need to solve the TPBVP, proceed as follows.

Claim B
The costate

$$\lambda^{(\mathrm{P})}(t) = 2P^{(\mathrm{P})}(t) \cdot x^*(t; x_0) \tag{55}$$

and the costate

$$\lambda^{(\mathrm{E})}(t) = 2P^{(\mathrm{E})}(t) \cdot x^*(t; x_0), \tag{56}$$

where $P^{(\mathrm{P})}(t)$ and $P^{(\mathrm{E})}(t)$ are real, symmetric $n \times n$ matrices $\forall\ 0 \le t \le T$.

Inserting (55) and (56) into (52)–(54) yields as set of two coupled Riccati type matrix differential equations

$$\frac{\mathrm{d}P^{(\mathrm{P})}}{\mathrm{d}t} = -A^T P^{(\mathrm{P})} - P^{(\mathrm{P})}A - Q^{(\mathrm{P})} + P^{(\mathrm{P})}B(R^{(\mathrm{P})})^{-1}B^T P^{(\mathrm{P})}$$
$$+\frac{1}{2}\left[P^{P)}C\left(R^{(\mathrm{E})}\right)^{-1}C^T P^{(\mathrm{E})} + P^{(\mathrm{E})}C^T\left(R^{(\mathrm{E})}\right)^{-1}CP^{(\mathrm{P})}\right],\ P^P(T) = Q_F^{(\mathrm{P})}$$

and

$$\frac{\mathrm{d}P^{(\mathrm{E})}}{\mathrm{d}t} = -A^T P^{(\mathrm{E})} - P^{(\mathrm{E})}A - Q^{(\mathrm{E})} + P^{(\mathrm{E})}C(R^{(\mathrm{E})})^{-1}C^T P^{(\mathrm{E})}$$
$$+\frac{1}{2}\left[P^{(\mathrm{E})}B(R^{(\mathrm{P})})^{-1}B^T P^{(\mathrm{P})} + P^{(\mathrm{P})}B^T\left(R^{(\mathrm{P})}\right)^{-1}BP^{(\mathrm{E})}\right],\ P^E(T) = -Q_F^{(\mathrm{E})}$$

Once $P^{(\mathrm{P})}(t)$ and $P^{(\mathrm{E})}(t)$ have been calculated, the open-loop NE strategy of P is explicitly given by

$$u^*(t; x_0) = -\frac{1}{2}(R^{(\mathrm{P})})^{-1}B^T P^{(\mathrm{P})}(t) \cdot x^*(t)$$

and the open-loop NE strategy of E is explicitly given by

$$v^*(t; x_0) = -\frac{1}{2}(R^{(\mathrm{E})})^{-1}C^T P^{(\mathrm{E})}(t) \cdot x^*(t);$$

the optimal trajectory $x^*(t)$ is given by the solution of the linear differential equation

$$\frac{\mathrm{d}x^*}{\mathrm{d}t} = \left\{ A - \frac{1}{2}\left[B(R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})} + C(R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})} \right] \right\} \cdot x^*,$$
$$x^*(0) = x_0, \ 0 \le t \le T$$

The above obtained result is summarized in

Theorem 2. *A (unique) solution to the open-loop nonzero-sum LQ differential game (16)–(18) exists* $\forall\ x_0 \in R^n$ *iff a solution on the interval* $0 \le t \le T$ *of the two coupled Riccati-type matrix differential equations*

$$\frac{\mathrm{d}P^{(\mathrm{P})}}{\mathrm{d}t} = A^T P^{(\mathrm{P})} + P^{(\mathrm{P})} A - P^{(\mathrm{P})} B (R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})} + Q^{(\mathrm{P})}$$
$$-\frac{1}{2}\left[P^{(\mathrm{P})} C \left(R^{(\mathrm{E})} \right)^{-1} C^T P^{(\mathrm{E})} + P^{(\mathrm{E})} C^T \left(R^{(\mathrm{E})} \right)^{-1} C P^{(\mathrm{P})} \right], \ P^P(0) = Q_F^{(\mathrm{P})} \tag{57}$$

and

$$\frac{\mathrm{d}P^{(\mathrm{E})}}{\mathrm{d}t} = A^T P^{(\mathrm{E})} + P^{(\mathrm{E})} A + Q^{(\mathrm{E})} - P^{(\mathrm{E})} C (R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})}$$
$$-\frac{1}{2}\left[P^{(\mathrm{E})} B \left(R^{(\mathrm{P})} \right)^{-1} B^T P^{(\mathrm{P})} + P^{(\mathrm{P})} B^T \left(R^{(\mathrm{P})} \right)^{-1} B P^{(\mathrm{E})} \right], \ P^E(T) = -Q_F^{(\mathrm{E})} \tag{58}$$

exists. The open-loop NE strategy of P is

$$u^*(t; x_0) = -\frac{1}{2}(R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})}(T-t) \cdot x^*(t) \tag{59}$$

and the open-loop NE strategy of E is

$$v^*(t; x_0) = -\frac{1}{2}(R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})}(T-t) \cdot x^*(t), \tag{60}$$

where $x^*(t)$*, the optimal trajectory, is given by the solution of the linear differential equation*

$$\frac{\mathrm{d}x^*}{\mathrm{d}t} = \left\{ A - \frac{1}{2}[B(R^{(\mathrm{P})})^{-1} B^T P^{(\mathrm{P})}(T-t) + C(R^{(\mathrm{E})})^{-1} C^T P^{(\mathrm{E})}(T-t)] \right\} \cdot x^*,$$
$$x^*(0) = x_0, \ \ 0 \le t \le T \tag{61}$$

Finally, the respective values of P and E are

$$V^{(\mathrm{P})}(x_0) = x_0^T P^{(\mathrm{P})}(T) x_0 \tag{62}$$

and

$$V^{(\mathrm{E})}(x_0) = x_0^T P^{(\mathrm{E})}(T) x_0 \tag{63}$$

Evidently, the solution of the open-loop nonzero-sum LQ differential game hinges on the solution of the set of Riccati equations (57) and (58) on the interval $0 \leq t \leq T$. A solution always exists for T sufficiently small.

5.1 Discussion

It is remarkable that also the solution of the open-loop LQ differential game hinges on the existence of a solution to a system of Riccati equations. However, the solutions $P^{(\mathrm{P})}(t)$ and $P^{(\mathrm{E})}(t)$ of Riccati differential equations (57) and (58) which pertain to the case where the players P and E both play open-loop are not the same as the solutions of Riccati differential equations (27) and (28) which pertain to the case where the players P and E both use closed-loop strategies. Thus, we denote the solution of the system of Riccati equations (27) and (28) by $P_{CC}^{(\mathrm{P})}(t)$, and $P_{CC}^{(\mathrm{E})}(t)$, and the solution of the system of Riccati equations (57) and (58) by $P_{OO}^{(\mathrm{P})}(t)$ and $P_{OO}^{(\mathrm{E})}(t)$. These determine the values of the respective closed-loop and open-loop nonzero-sum differential games.

Since Riccati equations (57) and (58) are not identical to Riccati equations (27) and (28), the open-loop values are different from the values obtained when feedback strategies are used. Indeed, consider the scalar minimum energy nonzero-sum differential game discussed in Sect. 3.1. In the scalar closed-loop game the Riccati equations, (27) and (28), are

$$\dot{P}_{CC}^{(\mathrm{P})} = 2aP_{CC}^{(\mathrm{P})} - \frac{b^2}{r^{(\mathrm{P})}}\left(P_{CC}^{(\mathrm{P})}\right)^2 - 2\frac{c^2}{r^{(\mathrm{E})}}P_{CC}^{(\mathrm{P})}P_{CC}^{(\mathrm{E})},\ P_{CC}^{(\mathrm{P})}(0) = q_F^{(\mathrm{P})}$$

$$\dot{P}_{CC}^{(\mathrm{E})} = 2aP_{CC}^{(\mathrm{E})} - \frac{c^2}{r^{(\mathrm{E})}}\left(P_{CC}^{(\mathrm{E})}\right)^2 - 2\frac{b^2}{r^{(\mathrm{P})}}P_{CC}^{(\mathrm{P})}P_{CC}^{(\mathrm{E})},\ P_{CC}^{(\mathrm{E})}(0) = -q_F^{(\mathrm{E})},\ 0 \leq t \leq T$$

We "integrate" the differential system and we calculate

$$P_{CC}^{(\mathrm{P})}(\Delta T) = q_F^{(\mathrm{P})} + \left[2aq_F^{(\mathrm{P})} - \frac{b^2}{r^{(\mathrm{P})}}\left(q_F^{(\mathrm{P})}\right)^2 + 2\frac{c^2}{r^{(\mathrm{E})}}q_F^{(\mathrm{P})}q_F^{(\mathrm{E})}\right]\cdot\Delta T$$

$$P_{CC}^{(\mathrm{E})}(\Delta T) = -\left\{q_F^{(\mathrm{E})} + \left[2aq_F^{(\mathrm{E})} + \frac{c^2}{r^{(\mathrm{E})}}\left(q_F^{(\mathrm{E})}\right)^2 - 2\frac{b^2}{r^{(\mathrm{P})}}q_F^{(\mathrm{P})}q_F^{(\mathrm{E})}\right]\cdot\Delta T\right\}$$

In the scalar open-loop game Riccati equations, (57) and (58), are

$$\dot{P}_{OO}^{(\mathrm{P})} = 2aP_{OO}^{(\mathrm{P})} - \frac{b^2}{r^{(\mathrm{P})}}\left(P_{OO}^{(\mathrm{P})}\right)^2 - \frac{c^2}{r^{(\mathrm{E})}}P_{OO}^{(\mathrm{P})}P_{OO}^{(\mathrm{E})},\ P_{OO}^{(\mathrm{P})}(0) = q_F^{(\mathrm{P})}$$

$$\dot{P}_{OO}^{(\mathrm{E})} = 2aP_{OO}^{(\mathrm{E})} - \frac{c^2}{r^{(\mathrm{E})}}\left(P_{OO}^{(\mathrm{E})}\right)^2 - \frac{b^2}{r^{(\mathrm{P})}}P_{OO}^{(\mathrm{P})}P_{OO}^{(\mathrm{E})},\ P_{OO}^{(\mathrm{E})}(0) = -q_F^{(\mathrm{E})},\ 0 \le t \le T$$

We "integrate" the differential system and we calculate

$$P_{OO}^{(\mathrm{P})}(\Delta T) = q_F^{(\mathrm{P})} + \left[2aq_F^{(\mathrm{P})} - \frac{b^2}{r^{(\mathrm{P})}}\left(q_F^{(\mathrm{P})}\right)^2 + \frac{c^2}{r^{(\mathrm{E})}}q_F^{(\mathrm{P})}q_F^{(\mathrm{E})}\right]\cdot\Delta T$$

$$P_{OO}^{(\mathrm{E})}(\Delta T) = -\left\{q_F^{(\mathrm{E})} + \left[2aq_F^{(\mathrm{E})} + \frac{c^2}{r^{(\mathrm{E})}}\left(q_F^{(\mathrm{E})}\right)^2 - \frac{b^2}{r^{(\mathrm{P})}}q_F^{(\mathrm{P})}q_F^{(\mathrm{E})}\right]\cdot\Delta T\right\}$$

From the above calculations we conclude

$$P_{OO}^{(\mathrm{P})}(\Delta T) < P_{CC}^{(\mathrm{P})}(\Delta T)$$

and

$$P_{OO}^{(\mathrm{E})}(\Delta T) < P_{CC}^{(\mathrm{E})}(\Delta T)$$

In summary, we have

Proposition 2. *In the scalar nonzero-sum LQ differential game and for T sufficiently small, the players' values satisfy*

$$P_{OO}^{(\mathrm{P})}(t) < P_{CC}^{(\mathrm{P})}(t)$$

and

$$P_{OO}^{(\mathrm{E})}(t) < P_{CC}^{(\mathrm{E})}(t)$$

$\forall\ 0 \le t \le T$.

It goes without saying that the range of the game horizon T s.t. Proposition 4 holds depends on the problem parameters a, b, c, $q_F^{(\mathrm{P})}$, $q_F^{(\mathrm{E})}$, $r^{(\mathrm{P})}$, and $r^{(\mathrm{E})}$.

Example 1. Scalar nonzero-sum differential game. The solutions $P_{CC}^{(\mathrm{P})}$ and $P_{CC}^{(\mathrm{E})}$ of Riccati equations (27) and (28) are compared to the solutions $P_{OO}^{(\mathrm{P})}$ and $P_{OO}^{(\mathrm{E})}$ of Riccati equations (57) and (58).

The parameters are $q_F^{(\mathrm{P})} = q_F^{(\mathrm{E})} = 1$, $r^{(\mathrm{P})} = \frac{1}{2}$, $r^{(\mathrm{E})} = 1$, $b = c = 1$, and $a = 1$. The values of the game when the initial state $\mid x_0 \mid = 1$ are shown in Fig. 1. The E-player is always better off when open-loop strategies are employed and the P-player is better off when open-loop strategies are employed, provided that the game horizon $T < 0.4258$.

When the dynamics parameter $a = -1$, both players are better off playing open-loop, irrespective of the length of the game horizon T—see, e.g., Fig. 2.

Not only does the open-loop solution *not* yield the solution to the closed-loop differential game, as is the case in optimal control and zero-sum differential games, but also, in addition, *both* players are better off using open-loop strategies as if only the initial state information x_0 were available to them. Indeed, it is most interesting that *both* players' open-loop values are lower than their respective closed-loop values: having access to the current state information does no good to the players.

6 Open-Loop vs. Closed-Loop Play in LQ Differential Games

Since in nonzero-sum differential games the open-loop solution $\neq$ closed-loop solution, it is interesting to also consider nonzero-sum differential games with an asymmetric information pattern where P uses a closed-loop strategy against player E who is committed to an open-loop strategy $v(t; x_0)$, $0 \leq t \leq T$, and vice versa.

We consider the nonzero-sum LQ differential game (16)–(18) where P uses state feedback whereas E plays open-loop. In this case player P uses DP against E's control $v(t)$, $0 \leq t \leq T$, whereas player E applies the PMM against P's state feedback strategy $u(t, x)$. Hence, the respective Hamiltonians of players P and E are as follows.

P's Hamiltonian

$$H^{(\mathrm{P})}(t, x, u, \lambda) = x^T Q^{(\mathrm{P})} x + u^T R^{(\mathrm{P})} u + \lambda^T [Ax + Bu + Cv(t)],$$

where

$$\lambda \equiv V_x^{(\mathrm{P})}$$

Consequently, the optimal "control law" is

$$u(t, x, V_x^{(\mathrm{P})}) = -\frac{1}{2}(R^{(\mathrm{P})})^{-1} B^T V_x^{(\mathrm{P})}$$

E's Hamiltonian

$$H^{(\mathrm{E})}(t, x, v, \lambda) = x^T Q^{(\mathrm{P})} x - v^T R^{(\mathrm{E})} v + \lambda^T [Ax + Bu + Cv(t)],$$

and consequently the optimal "control law" is

$$v(t, x, \lambda) = \frac{1}{2}(R^{(\mathrm{E})})^{-1} C^T \lambda$$

Thus, the following holds.

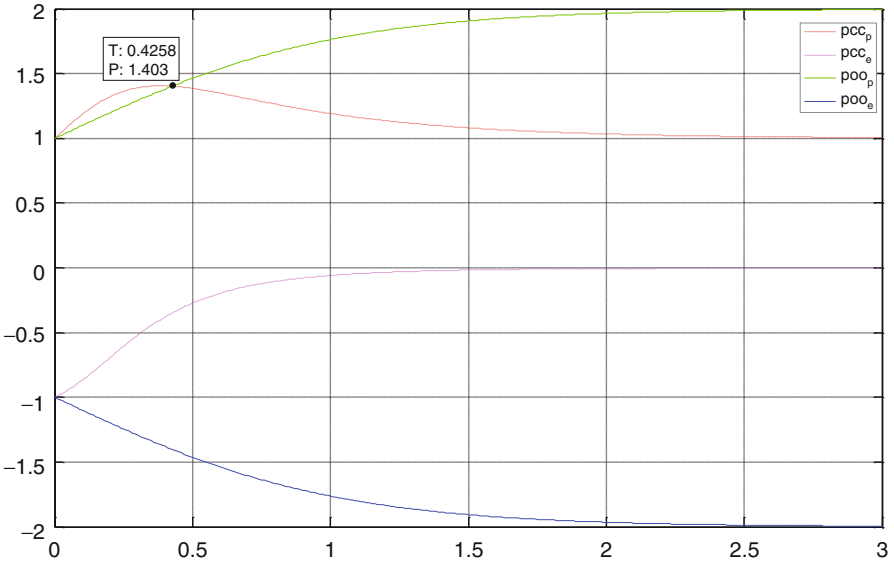

Fig. 1 Open-loop values < closed-loop values

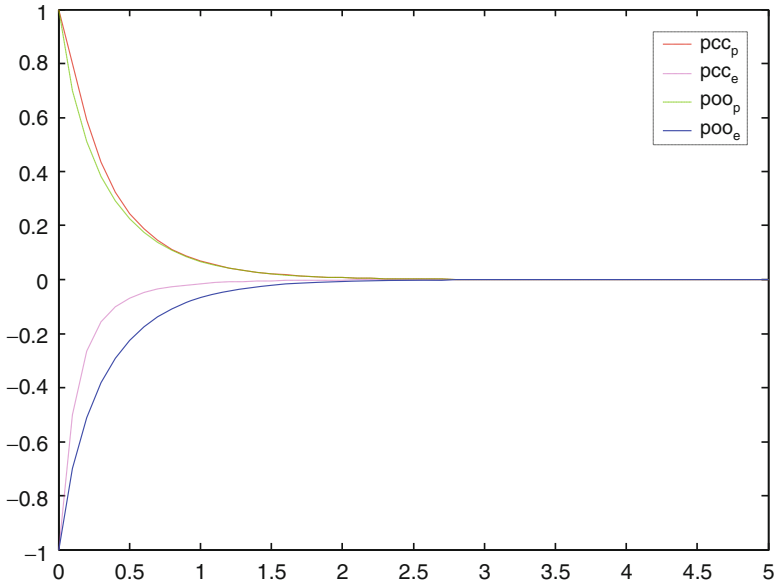

Fig. 2 Open-loop values < closed-loop values $\forall\ T > 0$

Applying the method of DP we obtain the PDE

$$-\frac{\partial V^{(\mathrm{P})}}{\partial t} = x^T Q^{(\mathrm{P})} x + x^T A^T V_x^{(\mathrm{P})} - \frac{1}{4}(V_x^{(\mathrm{P})})^T B(R^{(\mathrm{P})})^{-1} B^T V_x^{(\mathrm{P})} + \frac{1}{2}(V_x^{(\mathrm{P})})^T C(R^{(\mathrm{E})})^{-1} C^T \lambda,\ V_x^{(\mathrm{P})}(T,x) = x^T Q_F^{(\mathrm{P})} x\ \ \forall\ x \in R^n \quad (64)$$

Applying the PMM we obtain the costate equation

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -A^T\lambda - 2Q^{(\mathrm{E})}x - \left(\lambda^T B \frac{\partial u(t,x)}{\partial x}\right)^T,\ \lambda(T) = 2Q_F^{(\mathrm{E})}x(T)$$

Now

$$u(t,x) = -\frac{1}{2}(R^{(\mathrm{P})})^{-1} B^T V_x^{(\mathrm{P})}(t,x)$$

so that

$$\frac{\partial u(t,x)}{\partial x} = -\frac{1}{2}(R^{(\mathrm{P})})^{-1} B^T V_{xx}^{(\mathrm{P})}$$

and therefore

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -A^T\lambda - 2Q^{(\mathrm{E})}x - A^T x + \frac{1}{2}V_{xx}^{(\mathrm{P})}(t,x)B(R^{(\mathrm{P})})^{-1}B^T\lambda,$$
$$\lambda(T) = 2Q_F^{(\mathrm{E})}x(T) \quad (65)$$

Finally, the state evolves according to

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax - \frac{1}{2}B(R^{(\mathrm{P})})^{-1}B^T V_x^{(\mathrm{P})}(t,x) + \frac{1}{2}C(R^{(\mathrm{E})})^{-1}C^T\lambda,\ x(0) = x_0 \quad (66)$$

We must solve the above boundary value problem which entails a system of three equations—the PDE (64) and the two ODEs (65) and (66).

We shall require

Claim C

$$V^{(\mathrm{P})}(t,x) = x^T P^{(\mathrm{P})}(t)x,\ 0 \le t \le T, \quad (67)$$

where $P^{(\mathrm{P})}(t)$ is a real, symmetric matrix, $0 \le t \le T$.

Inserting the expression (67) for $V^{(\mathrm{P})}(t,x)$ into (64) and *symmetrizing* yield

$$-x^T\dot{P}^{(\mathrm{P})}x = x^T A^T P^{(\mathrm{P})}x + x^T P^{(\mathrm{P})}Ax - x^T P^{(\mathrm{P})}B(R^{(\mathrm{P})})^{-1}B^T P^{(\mathrm{P})}x + x^T Q^{(\mathrm{P})}x + \frac{1}{2}x^T P^{(\mathrm{P})}C(R^{(\mathrm{E})})^{-1}C^T\lambda + \frac{1}{2}\lambda^T C(R^{(\mathrm{E})})^{-1}C^T P^{(\mathrm{P})}x,$$
$$P^{(\mathrm{P})}(T) = Q_F^{(\mathrm{P})} \quad (68)$$

and inserting the expression (67) for $V^{(\mathrm{P})}(t,x)$ into (65) and (66) yields

$$\dot{\lambda} = -A^T\lambda - 2Q^{(\mathrm{E})}x + P^{(\mathrm{P})}B(R^{(\mathrm{P})})^{-1}B^T\lambda,\ \lambda(T) = 2Q_F^{(\mathrm{E})}x(T) \tag{69}$$

and

$$\dot{x} = Ax - B(R^{(\mathrm{P})})^{-1}B^TP^{(\mathrm{P})}x + \frac{1}{2}C(R^{(\mathrm{E})})^{-1}C^T\lambda,\ x(0) = x_0 \tag{70}$$

We shall also require

Claim D

$$\lambda(t) = -2P^{(\mathrm{E})}(t)x, \tag{71}$$

where $P^{(\mathrm{E})}(t)$ is a real, symmetric matrix, $0 \le t \le T$.

Inserting the expression (71) for $\lambda(t)$ into (68) yields

$$\begin{aligned} -\dot{P}^{(\mathrm{P})} &= A^TP^{(\mathrm{P})} + P^{(\mathrm{P})}A - P^{(\mathrm{P})}B(R^{(\mathrm{P})})^{-1}B^TP^{(\mathrm{P})} + Q^{(\mathrm{P})} \\ &\quad -P^{(\mathrm{P})}C(R^{(\mathrm{E})})^{-1}C^TP^{(\mathrm{E})} - P^{(\mathrm{E})}C(R^{(\mathrm{E})})^{-1}C^TP^{(\mathrm{P})}x,\ P^{(\mathrm{P})}(T) = Q_F^{(\mathrm{P})} \end{aligned} \tag{72}$$

Furthermore, differentiation of (71) yields

$$-\dot{\lambda} = 2\dot{P}^{(\mathrm{E})}x + 2P^{(\mathrm{E})}\dot{x} \tag{73}$$

Inserting (70) into (72) and using Ansatz E yield

$$\begin{aligned} -\dot{\lambda} &= 2\dot{P}^{(\mathrm{E})}x + 2P^{(\mathrm{E})}\left[Ax - B(R^{(\mathrm{P})})^{-1}B^TP^{(\mathrm{P})}x + \frac{1}{2}C(R^{(\mathrm{E})})^{-1}C^T\lambda\right] \\ &= 2\dot{P}^{(\mathrm{E})}x + 2P^{(\mathrm{E})}\left[Ax - B(R^{(\mathrm{P})})^{-1}B^TP^{(\mathrm{P})}x - C(R^{(\mathrm{E})})^{-1}C^TP^{(\mathrm{E})}x\right] \end{aligned} \tag{74}$$

Next, inserting the expression (74) for $\dot{\lambda}$ into (69) and reusing Ansatz E yield

$$\begin{aligned} -\dot{P}^{(\mathrm{E})} &= A^TP^{(\mathrm{E})} + P^{(\mathrm{E})}A - P^{(\mathrm{E})}C(R^{(\mathrm{E})})^{-1}C^TP^{(\mathrm{E})} - Q^{(\mathrm{E})} \\ &\quad -P^{(\mathrm{E})}B(R^{(\mathrm{P})})^{-1}B^TP^{(\mathrm{P})} - P^{(\mathrm{P})}B(R^{(\mathrm{P})})^{-1}B^TP^{(\mathrm{E})},\ P^{(\mathrm{E})}(T) = Q_F^{(\mathrm{E})} \end{aligned} \tag{75}$$

We have obtained a system of coupled Riccati equations, (72) and (75), which is identical to (27) and (28). The solution of the system of Riccati equations (27) and (28) yields the optimal strategies

$$u^*(t,x) = -(R^{(\mathrm{P})})^{-1}B^TP^{(\mathrm{P})}(t)x$$

and

$$v^*(t;x_0) = (R^{(\mathrm{E})})^{-1}C^T P^{(\mathrm{E})}(t)x^*(t)$$

Having obtained the solution of Riccati systems (27) and (28), the optimal trajectory $x^*(t)$ is given by the solution of the linear differential equation

$$\dot{x}^* = \left[A - B(R^{(\mathrm{P})})^{-1}B^T P^{(\mathrm{P})}(t) + C(R^{(\mathrm{E})})^{-1}C^T P^{(\mathrm{E})}(t)\right]x^*,\ x^*(0) = x_0$$

Remark 1. The above alluded to symmetrization step yields symmetric Riccati equations (72) and (75), for otherwise "new" Riccati equations are obtained, as in [9].

We shall use the following notation. In the game where P plays closed-loop and E plays open-loop we denote the solution of the system of Riccati equations (27) and (28) by $P_{CO}^{(\mathrm{P})}(t)$ and $P_{CO}^{(\mathrm{E})}(t)$. Conversely, if P plays open-loop and E plays closed-loop, the system of Riccati equations (27) and (28) is re-derived and its solution is then denoted by $P_{OC}^{(\mathrm{P})}(t)$ and $P_{OC}^{(\mathrm{E})}(t)$. In both cases, the system of Riccati equations is the system (27) and (28) which pertains to the case where *both* players use closed-loop strategies and the solution $P^{(\mathrm{P})}(t)$ and $P^{(\mathrm{E})}(t)$ of (27) and (28) is denoted by $P_{CC}^{(\mathrm{P})}(t)$ and $P_{CC}^{(\mathrm{E})}(t)$, respectively.

In summary, the following holds.

Proposition 3. *In nonzero-sum open-loop/closed-loop and closed-loop/open-loop LQ differential games, the players' values are equal to the players' values in the game where both players use closed-loop strategies—in other words,*

$$P_{OC}^{(\mathrm{P})}(t) = P_{CC}^{(\mathrm{P})}(t) = P^{(\mathrm{P})}(t)$$

and

$$P_{OC}^{(\mathrm{E})}(t) = P_{CC}^{(\mathrm{E})}(t) = P^{(\mathrm{E})}(t)$$

Similarly,

$$P_{CO}^{(\mathrm{P})}(t) = P_{CC}^{(\mathrm{P})}(t) = P^{(\mathrm{P})}(t)$$

and

$$P_{CO}^{(\mathrm{E})}(t) = P_{CC}^{(\mathrm{E})}(t) = P^{(\mathrm{E})}(t)$$

$\forall\ 0 \le t \le T$.

Also recall that for the case where both P and E play open-loop, the solution of the Riccati system (57) and (58), denoted by $P_{OO}^{(\mathrm{P})}(t)$ and $P_{OO}^{(\mathrm{E})}(t)$, applies.

6.1 Discussion

When P plays closed-loop, the players' values are as in the closed-loop differential game where both players use closed-loop strategies, that is, $V^{(\mathrm{P})}(t,x) = x^T P_{CC}^{(\mathrm{P})}(t)x = x^T P^{(\mathrm{P})}(t)x$ and $V^{(\mathrm{E})}(t,x) = x^T P_{CC}^{(\mathrm{E})}(t)x = x^T P^{(\mathrm{E})}(t)x$—irrespective of whether player E plays open-loop or closed-loop. The converse is also true: when E plays closed-loop, the players' values are as in the closed-loop differential game, irrespective of whether player P plays open-loop or closed-loop. However, it is advantageous for *both* players to play open-loop; their values/costs are reduced compared to the case where they both play closed-loop:

$$P_{OO}^{(\mathrm{P})}(t) < P_{CC}^{(\mathrm{P})}(t)$$

and

$$P_{OO}^{(\mathrm{E})}(t) < P_{CC}^{(\mathrm{E})}(t)$$

If however just one of the players plays open-loop and his opponent plays closed-loop, then both players' values are the higher closed-loop values.

7 Conclusion

In this article nonzero-sum differential games are addressed. When nonzero-sum games are considered, there is no reason to believe that strategies might exist s.t. all the players are able to minimize their own cost and the natural optimality concept is the Nash equilibrium (NE). Now, the NE concept is somewhat problematic, because, first of all, it is not necessarily unique. If the NE *is* unique, the definition of NE strategy is appealing: by definition, a player's NE strategy is s.t. should he not adhere to it while his opposition does stick to its NE strategy, his cost will increase and he will be penalized. Thus, assuming that the opposition will in fact execute its NE strategy, a player will be wise to adhere to his NE strategy. Note however that this is not to say that by having all parties deviate from their respective NE strategies, they would not all do better—think of the Prisoner's Dilemma game. Now, in the special case of zero-sum games, the NE is a saddle point: hence, an NE strategy is a security strategy, the value of the game is unique, and the uniqueness of optimal strategies is not an issue because they are interchangeable. Evidently, nonzero-sum games are more complex than zero-sum games. This is even more so when dynamic games, and, in particular, nonzero-sum differential games, are addressed. In this article the open-loop and closed-loop information patterns in nonzero-sum differential games are examined. The results are specialized with reference to LQ differential games. It is explicitly shown that in LQ differential games, somewhat paradoxically, open-loop NE strategies are superior to closed-loop NE strategies. Moreover, even if only

one party employs closed-loop control, both players' values are the inferior values of the game where both players employ closed-loop strategies. This state of affairs can be attributed to the inherent weakness of the NE solution concept, which is not apparent in zero-sum differential games.

References

1. Case, J.H.: Equilibrium Points of N-Person Differential Games, University of Michigan, Department of industrial Engineering, TR No 1967–1, (1967)
2. Case, J.H.: Toward a theory of many player differential games. SIAM J. Control **7**(2), 179–197 (1969)
3. Starr, A.W., Ho, Y.C.: Nonzero-sum differential games. J. Optimiz. Theory. App. **3**(3), 184–206 (1969)
4. Starr, A.W., Ho, Y.C.: Further properties of nonzero-sum differential games. J. Optimiz. Theory. App. **3**(4), 207–218 (1969)
5. Byung-Wook Wie: A differential game model of nash equilibrium on a congested traffic network. Networks **23**(6), 557–565 (1993)
6. Byung-Wook Wie: A differential game approach to the dynamics of mixed behavior traffic network equilibrium problem. Eur. J. Oper. Res. **83**, 117–136 (1995)
7. Olsder, G.J.: On open and closed-loop bang-bang control in nonzero-sum differential games. SIAM J. Control Optim. **40**(4), 1087–1106 (2002)
8. Basar, T., Olsder G.J.: Dynamic Noncooperative Game Theory. SIAM, London, UK (1999)
9. Engwerda, J.C: LQ Dynamic Optimization and Differential Games. Wiley, Chichester, UK (2005)

Information Considerations in Multi-Person Cooperative Control/Decision Problems: Information Sets, Sufficient Information Flows, and Risk-Averse Decision Rules for Performance Robustness

Khanh D. Pham and Meir Pachter

Abstract The purpose of this research investigation is to describe the main concepts, ideas, and operating principles of stochastic multi-agent control or decision problems. In such problems, there may be more than one controller/agent not only trying to influence the continuous-time evolution of the overall process of the system, but also being coupled through the cooperative goal for collective performance. The mathematical treatment is rather fundamental; the emphasis of the article is on motivation for using the common knowledge of a process and goal information. The article then starts with a discussion of sufficient information flows with a feedforward structure providing coupling information about the control/decision rules to all agents in the cooperative system. Some attention has been paid to the design of decentralized filtering via constrained filters for the multi-agent dynamic control/decision problem considered herein. The main focus is on the synthesis of decision strategies for reliable performance. That is on mathematical statistics associated with an integral-quadratic performance measure of the generalized chi-squared type, which can later be exploited as the essential commodity to ensure much of the design-in reliability incorporated in the development phase. The last part of the article gives a comprehensive presentation of the broad and still developing area of risk-averse controls. It is possible to show that each agent with

K.D. Pham
Air Force Research Laboratory, Space Vehicles Directorate,
Kirtland Air Force Base, Albuquerque, NM 87117, USA
e-mail: AFRL.RVSV@kirtland.af.mil

M. Pachter (✉)
Air Force Institute of Technology, Department of Electrical and Computer Engineering,
Wright-Patterson Air Force Base, Dayton, OH 45433, USA
e-mail: meir.pachet@afit.edu

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*,
Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_16,

risk-averse attitudes not only adopts the use of a full-state dimension and linear dynamic compensator driven by local measurements, but also generates cooperative control signals and coordinated decisions.

Keywords Stochastic multi-agent cooperative control/decision problems • Performance-measure statistics • Performance reliability • Risk-averse control decisions • Matrix minimum principle • Necessary and sufficient conditions

1 Introduction

Throughout the article, the superscript T in the notation is denoted for the transposition of vector or matrix entities. In addition, $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ is a complete filtered probability space and a standard p-dimensional Wiener process $w(t) \equiv w(t, \omega)$ and $\omega \in \Omega$ with the correlation of independent increments given by $E\{[w(t_1) - w(t_2)][w(t_1) - w(t_2)]^{\mathrm{T}}\} = W|t_1 - t_2|$, $W > 0$ for all $t_1, t_2 \in [0, T]$ and $w(0) = 0$ which generates the filtration $\mathbb{F} \triangleq \mathcal{F}_t$ and $\mathcal{F}_t = \sigma\{w(s) : 0 \leq s \leq t\}$ augmented by all $\mathbb{P}$-null sets in $\mathcal{F}$. Consider the following controlled stochastic problem:

$$\begin{aligned} \mathrm{d}x(t) &= f(t, x(t), u(t))\mathrm{d}t + g(t)\mathrm{d}w(t), \qquad t \in [0, T] \\ x(0) &= x_0 \end{aligned} \tag{1}$$

and a performance measure

$$J(u(\cdot)) = \int_0^T q(t, x(t), u(t))\mathrm{d}t + h(x(T)) \,. \tag{2}$$

Here, $x(t) \equiv x(t, \omega)$ is the controlled state process valued in $\mathbb{R}^n$, $u(t) \equiv u(t, \omega)$ is the control process valued in some set $U \subset \mathbb{R}^m$ bounded or unbounded and $g(t) \equiv g(t, \omega)$ is an $n \times p$ matrix, $\omega \in \Omega$. In the setting (1)–(2), $f(t, x, u, \omega) : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \Omega \mapsto \mathbb{R}^n$, $q(t, x, u, \omega) : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \Omega \mapsto \mathbb{R}$ and $h(x) : \mathbb{R}^n \mapsto \mathbb{R}$ are given measurable functions.

It is assumed that the random functions $f(t, x, u, \omega)$ and $q(t, x, u, \omega)$ are continuous for fixed $\omega \in \Omega$ and are progressively measurable with respect to $\mathcal{F}_t$ for fixed (x, u). The function $h(x) : \mathbb{R}^n \mapsto \mathbb{R}$ is continuous.

To best explain the sort of applications to be addressed for the stochastic control problem (1)–(2), it is commenced by giving a brief description of stochastic multi-agent cooperative decision and control problems of more than one controllers and/or agents, who not only try to influence the continuous-time evolution of the overall controlled process with local imperfect measurements but also coordinate actions through the same performance measure. In such a partially decentralized situation, $u(\cdot) \triangleq (u_1(\cdot), \ldots, u_N(\cdot))$ of which $u_i(\cdot)$ is the extreme control of the ith controller or agent, valued in $U_i \subset \mathbb{R}^{m_i}$ with $m_1 + \cdots + m_N = m$, and is to be chosen to

optimize expected values and variations of $J(u(\cdot))$. Furthermore, it seems unlikely that a closed-loop solution will be available in closed-form for this stochastic multi-agent cooperative control and/or decision problem except, possibly, under the structural constraints of linear system dynamics, quadratic cost functionals, and additive independent white Gaussian noises corrupting the system dynamics and measurements.

For this reason, attention in this research investigation is directed primarily toward the stochastic control and/or decision problem with multiple agents, which has linear system dynamics, quadratic cost functionals, and uncorrelated standard Wiener process noises additively corrupting the controlled system dynamics and output measurements. Notice that under these conditions the quadratic cost functional associated with this problem class is a random variable with the generalized chi-squared probability distributions. If a measure of uncertainty, such as the variance of the possible outcome, was used in addition to the expected outcome, then the agents should be able to correctly order preferences for alternatives. This claim seems plausible, but it is not always correct. Various investigations have indicated that any evaluation scheme based on just the expected outcome and outcome variations would necessarily imply indifference between some courses of action; therefore, no criterion based solely on the two attributes of means and variances can correctly represent their preferences. See [1, 2] for early important observations and findings.

As will be clear in the research development herein, the shape and functional form of a utility function tell us a great deal about the basic attitudes of the agents or decision makers toward the uncertain outcomes or performance risks. Of particular interest, the new utility function or the so-called risk-value aware performance index, which is proposed herein as a linear manifold defined by a finite number of centralized moments associated with a random performance measure of integral quadratic form, will provide a convenient allocation representation of apportioning performance robustness and reliability requirements into the multi-attribute requirement of qualitative characteristics of expected performance and performance risks. The technical approach to a solution for the stochastic multi-agent cooperative control or decision problem under consideration and its research contributions rest upon: (a) the acquisition and utilization of insights, regarding whether the agents are risk averse or risk prone and thus restrictions on the utility functions implied by these attitudes and (b) the adaptation of decision strategies to meet difficult environments, as well as best courses of action to ensure performance robustness and reliability, provided that the agents be subscribed to certain attitudes.

The rest of the article is organized as follows. In Sect. 2 the stochastic two-agent cooperative problem with the linear-quadratic structure is formulated. Section 3 is devoted to decentralized filtering via constrained filters derived for cooperative agents with different information patterns. In addition, the mathematical analysis of higher-order statistics associated with the performance-measure is of concern in Sect. 4. Section 5 applies the first- and second-order conditions of the matrix minimum principle for optimal controls of the stochastic cooperative problem. Finally, Sect. 6 gives some concluding remarks.

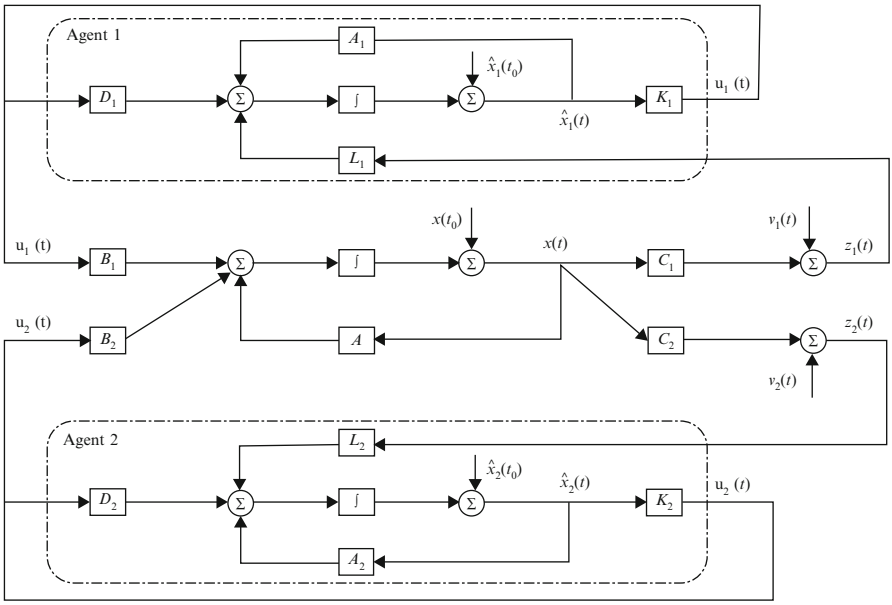

Fig. 1 Interaction architecture of a stochastic multi-agent cooperative system

2 Problem Formulation and Preliminaries

In this section, some preliminaries are in order. First of all, some spaces of random variables and stochastic processes are introduced

$$L^2_{\mathcal{F}_t}(\Omega;\mathbb{R}^n) \triangleq \{\eta : \Omega \mapsto \mathbb{R}^n | \eta \text{ is } \mathcal{F}_t\text{-measurable}, E\{||\eta||^2\} < \infty\},$$

$$L^2_{\mathcal{F}}(0,T;\mathbb{R}^k) \triangleq \left\{ f : [0,T] \times \Omega \mapsto \mathbb{R}^k | f(\cdot) \text{ is } \mathbb{F}\text{-adapted}, \right.$$

$$\left. E\left\{\int_0^T ||f(t)||^2 \mathrm{d}t\right\} < \infty\right\}.$$

Next, as a special case of the setting (1)–(2) it is of interest to consider a problem class of stochastic two-agent control and decision systems like those shown in Fig. 1 by use of the structural choice, referred to as "linear system dynamics and output measurements." It shall be stressed that the stochastic control or decision problem concerned is to be influenced by two cooperative agents as described by the controlled stochastic differential equation

$$\mathrm{d}x(t) = (A(t)x(t) + B_1(t)u_1(t) + B_2(t)u_2(t))\,\mathrm{d}t + G(t)\mathrm{d}w(t), \quad x(0) = x_0 \quad (3)$$

to which agent 1, controlling u_1, has available measurements of the form

$$\mathrm{d}y_1(t) = C_1(t)x(t)\mathrm{d}t + \mathrm{d}v_1(t)\,, \tag{4}$$

while agent 2, controlling u_2, has available measurements

$$\mathrm{d}y_2(t) = C_2(t)x(t)\mathrm{d}t + \mathrm{d}v_2(t)\,. \tag{5}$$

Here $x(t) \equiv x(t,\omega)$ is the controlled state process valued in $\mathbb{R}^n$; $u_1(t) \equiv u_1(t,\omega)$ and $u_2(t) \equiv u_2(t,\omega)$ are the control processes valued in some admissible sets $U_1 \subseteq L^2_{\mathcal{F}}(0,T;\mathbb{R}^{m_1})$ and $U_2 \subseteq L^2_{\mathcal{F}}(0,T;\mathbb{R}^{m_2})$; and $y_1(t) \equiv y_1(t,\omega)$ and $y_2(t) \equiv y_2(t,\omega)$ are the measured outputs of the manipulated process $x(t)$.

In the problem description, as described by equations (3)–(5), the system coefficients $A(t) \equiv A(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{n\times n}$, $B_1(t) \equiv B_1(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{n\times m_1}$, $B_2 \equiv B_2(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{n\times m_2}$, $G(t) \equiv G(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{n\times p}$, as well as the measurement coefficients $C_1(t) \equiv C_1(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{r_1\times n}$ and $C_2 \equiv C_2(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{r_2\times n}$, are continuous-time matrix functions. The random measurement disturbances $v_i(t) \equiv v_i(t,\omega)$ for $i = 1,2$ are assumed to be the uncorrelated stationary Wiener processes with the correlations of independent increments for all $t_1, t_2 \in [0,T]$

$$E\left\{[v_1(t_1) - v_1(t_2)][v_1(t_1) - v_1(t_2)]^{\mathrm{T}}\right\} = V_1|t_1 - t_2|\,, \quad V_1 > 0$$
$$E\left\{[v_2(t_1) - v_2(t_2)][v_2(t_1) - v_2(t_2)]^{\mathrm{T}}\right\} = V_2|t_1 - t_2|\,, \quad V_2 > 0$$

whose a priori second-order statistics V_1 and V_2 are also assumed known to both agents. For simplicity, both agents consider the initial state $x(0)$ to be known.

Associated with each $(u_1,u_2) \in U_1 \times U_2$ is a path-wise finite-horizon integral-quadratic form (IQF) performance measure with the generalized chi-squared random distribution, for which both agents attempt to coordinate their actions

$$J(u_1(\cdot),u_2(\cdot)) = x^T(T)Q_Tx(T) + \int_0^T \left[x^T(t)Q(t)x(t) + u_1^T(t)R_1(t)u_1(t) + u_2^T(t)R_2(t)u_2(t)\right]\mathrm{d}t \tag{6}$$

where the terminal penalty weighting $Q_T \in \mathbb{R}^{n\times n}$, the state weighting $Q(t) \equiv Q(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{n\times n}$ and control weightings $R_1(t) \equiv R_1(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{m_1\times m_1}$ and $R_2(t) \equiv R_2(t,\omega) : [0,T]\times\Omega \mapsto \mathbb{R}^{m_2\times m_2}$ are continuous-time matrix functions with the properties of symmetry and positive semi-definiteness. In addition, $R_1(t)$ and $R_2(t)$ are invertible.

Denote by Y_i and $i = 1,2$ the output functions measured by agents up to time t

$$Y_i(t) \triangleq \{(\tau, y_i(\tau));\ \tau \in [0,t)\}\,, \qquad i = 1,2\,.$$

Then the information structure is defined as follows:

$$Z_i(t) \triangleq Y_i(t) \cup \{\text{a priori information}\}\,.$$

Notice that the information structures $Z_i(t)$ and $i = 1,2$ include the a priori information available to agents so that in particular, either $Z_i(0)$ is simply the a priori information when either of them has no output measurements at all.

3 Decentralized Filtering via Constrained Filters

This work is concerned with decentralized filtering where each agent is constrained to make local noise state measurements. No exchange of online information is allowed, and each agent must generate its own online control decisions based upon its local processing resources. As an illustration, the information describing the system of the form (3)–(5) can be naturally grouped into three classes: (i) local model data, $I_{MD_i}(t) \triangleq \{A(t), B_i(t), G(t), C_i(t)\}$; (ii) local process data of statistical parameters concerning the stochastic processes, $I_{PD_i} \triangleq \{x_0, W, V_i\}$; and (iii) online data available from local measurements, $I_{OD_i}(t) \triangleq \{y_i(t)\}$ with $i = 1,2$ and $t \in [0,T]$. Hence, the information flow, as defined by $I_i(t) \triangleq I_{MD_i}(t) \cup I_{PD_i} \cup I_{OD_i}(t)$, is available at agent i's location.

Next, a simple class of implementable filter structures is introduced for the case of distributed information flows herein. Subsequently, instead of allowing each agent to preserve and use the entire output function that it has measured up to the time t, agent i is now restricted to generate and use only a vector-valued function that satisfies a linear, nth order dynamic system, which also gives unbiased estimates

$$\mathrm{d}\hat{x}_i(t) = (A_i(t)\hat{x}_i(t) + D_i(t)u_i(t))\mathrm{d}t + L_i(t)\mathrm{d}y_i(t)\,, \qquad i = 1,2 \tag{7}$$

wherein the continuous-time matrices $A_i(t) \equiv A_i(t,\omega) : [0,T] \times \Omega \mapsto \mathbb{R}^{n\times n}$, $D_i(t) \equiv D_i(t,\omega) : [0,T] \times \Omega \mapsto \mathbb{R}^{n\times m_i}$, $L_i(t) \equiv L_i(t,\omega) : [0,T] \times \Omega \mapsto \mathbb{R}^{n\times r_i}$ and the initial condition $\hat{x}_i(0)$ are to be selected by agent i while the expected value of the current state $x(t)$ given the measured output function $Y_i(t)$ is denoted by $\hat{x}_i(t)$. Notice that the filter (7), as proposed here is not the Stratonovich–Kalman–Bucy filter although it is linear and unbiased. The decentralized filtering problem is to determine matrix coefficients $A_i(\cdot)$, $D_i(\cdot)$, $L_i(\cdot)$, and initial filter states such that $\hat{x}_i(t)$ are unbiased estimates of $x(t)$ for all $u_i(t)$ and $i = 1,2$.

With respect to the structures of online dynamic control as considered in Fig. 1, practical control decisions with feedback $u_i = u_i(t, Z_i(t))$ for agent i and $i = 1,2$ are reasonably constrained to be linear transformations of unbiased estimates from the linear filters driven by the local measurements (7)

$$u_i(t) = u_i(t, \hat{x}_i(t)) \triangleq K_i(t)\hat{x}_i(t)\,, \qquad i = 1,2 \tag{8}$$

where the decision control gains $K_i(t) = K_i(t,\omega) : [0,T] \times \Omega \mapsto \mathbb{R}^{m_i \times n}$ will be appropriately defined such that the corresponding set U_i of admissible controls consisting of all functions $u_i(t,\omega) : [0,T] \times \Omega \mapsto \mathbb{R}^{m_i}$, which are progressively measurable with respect to $\mathcal{F}_t$ and $E\{\int_0^T ||u_i(t,\omega)||^2 \mathrm{d}t\} < \infty$.

Using (3), (7), and (8) it is easily shown that the estimation errors satisfy

$$\begin{aligned} \mathrm{d}x(t) - \mathrm{d}\hat{x}_1(t) = {} & (A(t) - A_1(t) + B_2(t)K_2(t) - L_1(t)C_1(t))x(t)\mathrm{d}t \\ & - B_2(t)K_2(t)(x(t) - \hat{x}_2(t))\mathrm{d}t + A_1(t)(x(t) - \hat{x}_1(t))\mathrm{d}t \\ & + (B_1(t) - D_1(t))K_1(t)\hat{x}_1(t)\mathrm{d}t + G(t)\mathrm{d}w(t) - L_1(t)\mathrm{d}v_1(t) \end{aligned} \tag{9}$$

$$\begin{aligned} \mathrm{d}x(t) - \mathrm{d}\hat{x}_2(t) = {} & (A(t) - A_2(t) + B_1(t)K_1(t) - L_2(t)C_2(t))x(t)\mathrm{d}t \\ & - B_1(t)K_1(t)(x(t) - \hat{x}_1(t))\mathrm{d}t + A_2(t)(x(t) - \hat{x}_2(t))\mathrm{d}t \\ & + (B_2(t) - D_2(t))K_2(t)\hat{x}_2(t)\mathrm{d}t + G(t)\mathrm{d}w(t) - L_2(t)\mathrm{d}v_2(t)\,. \end{aligned} \tag{10}$$

Furthermore, it requires to have both $\hat{x}_1(t)$ and $\hat{x}_2(t)$ to be unbiased estimates of $x(t)$ for all $u_1(t)$ and $u_2(t)$, that is, for all $t \in [0,T]$, $i = 1,2$ and $j = 1,2$

$$E\{x(t) - \hat{x}_i(t)|Z_j(t)\} = 0. \tag{11}$$

Now if the requirement (11) is satisfied, then for each t it follows that

$$E\{\mathrm{d}x(t) - \mathrm{d}\hat{x}_i(t)|Z_j(t)\} = 0\,, \qquad i = 1,2\,, \qquad j = 1,2. \tag{12}$$

Hence, from (9), (10), and the fact that $w(t)$ and $v_i(t)$ with $i = 1,2$ are the zero-mean random processes the necessary conditions for unbiased estimates then are

$$A_1(t) = A(t) + B_2(t)K_2(t) - L_1(t)C_1(t) \tag{13}$$

$$A_2(t) = A(t) + B_1(t)K_1(t) - L_2(t)C_2(t) \tag{14}$$

$$D_1(t) = B_1(t) \tag{15}$$

$$D_2(t) = B_2(t)\,. \tag{16}$$

In addition, letting $t = 0$ in (11) results in the condition

$$\hat{x}_i(0,\omega) = x_0, \qquad \forall \omega \in \Omega, \qquad i = 1,2. \tag{17}$$

On the other hand, using conditions (13)–(17) together with expressions (9) and (10) it follows that for $t \in [0,T]$ and $j = 1,2$

$$E\{\mathrm{d}x(t) - \mathrm{d}\hat{x}_1(t)|Z_j(t)\}$$
$$= A_1(t)E\{x(t) - \hat{x}_1(t)|Z_j(t)\}\mathrm{d}t - B_2(t)K_2(t)E\{x(t) - \hat{x}_2(t)|Z_j(t)\}\mathrm{d}t$$

$$E\{\mathrm{d}x(t) - \mathrm{d}\hat{x}_2(t)|Z_j(t)\}$$
$$= A_2(t)E\{x(t) - \hat{x}_2(t)|Z_j(t)\}\mathrm{d}t - B_1(t)K_1(t)E\{x(t) - \hat{x}_1(t)|Z_j(t)\}\mathrm{d}t$$

and

$$E\{x(t) - \hat{x}(t)|Z_j(t)\}\big|_{t=0} = 0\,.$$

Therefore, the conditions (13)–(17) are also sufficient for unbiased estimates. Henceforth, the class of decentralized filtering via constrained filters is characterized by the stochastic differential equation together with $i = 1, 2$, $j = 1, 2$, and $j \neq i$

$$\mathrm{d}\hat{x}_i(t) = [(A(t) + B_j(t)K_j(t))\hat{x}_i(t) + B_i(t)u_i(t)]\mathrm{d}t + L_i(t)(\mathrm{d}y_i - C_i(t)\hat{x}_i(t))$$
$$\hat{x}_i(0) = x_0 \tag{18}$$

wherein the filter gain $L_i(t)$ remains to be chosen by agent i in an optimal manner relative to the collective performance measure defined in (6).

4 Mathematical Statistics for Collective Performance Robustness

To progress toward the cooperation situation, the aggregate dynamics composing of agent interactions, distributed decision making and local autonomy are, therefore, governed by the controlled stochastic differential equation

$$\mathrm{d}z(t) = \hat{F}(t)z(t)\mathrm{d}t + \hat{G}(t)\mathrm{d}\hat{w}(t)\,, \qquad z(0) = z_0 \tag{19}$$

in which for each $t \in [0, T]$, the augmented state variables, the underlying process noises, and the system coefficients are given by

$$z \triangleq \begin{bmatrix} x \\ x - \hat{x}_1 \\ x - \hat{x}_2 \end{bmatrix}, \quad z_0 \triangleq \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix}, \quad \hat{w} \triangleq \begin{bmatrix} w \\ v_1 \\ v_2 \end{bmatrix}$$

and

$$\hat{F} \triangleq \begin{bmatrix} A + B_1K_1 + B_2K_2 & -B_1K_1 & -B_2K_2 \\ 0 & A + B_2K_2 - L_1C_1 & -B_2K_2 \\ 0 & -B_1K_1 & A + B_1K_1 - L_2C_2 \end{bmatrix} \tag{20}$$

$$\hat{G} \triangleq \begin{bmatrix} G & 0 & 0 \\ G & -L_1 & 0 \\ G & 0 & -L_2 \end{bmatrix}, \quad \hat{W} \triangleq \begin{bmatrix} W & 0 & 0 \\ 0 & V_1 & 0 \\ 0 & 0 & V_2 \end{bmatrix} \tag{21}$$

with $E\{[\hat{w}(t_1) - \hat{w}(t_2)][\hat{w}(t_1) - \hat{w}(t_2)]^{\mathrm{T}}\} = \hat{W}|t_1 - t_2|$ for all $t_1, t_2 \in [0, T]$.

Moreover, the pairs $(A(\cdot), B_i(\cdot))$ and $(A(\cdot), C_i(\cdot))$ for $i = 1, 2$ are assumed to be uniformly stabilizable and detectable, respectively. Under this assumption, such feedback and filter gains $K_i(\cdot)$ and $L_i(\cdot)$ for $i = 1, 2$ exist so that the aggregate system dynamics is uniformly exponentially stable. In other words, there exist positive constants η_1 and η_2 such that the pointwise matrix norm of the state transition matrix of the closed-loop system matrix $\hat{F}(\cdot)$ satisfies the inequality

$$||\Phi_{\hat{F}}(t, \tau)|| \leq \eta_1 e^{-\eta_2(t-\tau)}, \qquad \forall\, t \geq \tau \geq 0\,. \tag{22}$$

In most of the type of problems under consideration and available results in team theory [3] and large-scale systems [4], it is apparent that there is lack of analysis of performance risk and stochastic preferences beyond statistical averaging. Henceforth, the following development is vital to examine what it means for performance riskiness from the standpoint of higher-order characteristics pertaining to performance sampling distributions. Specifically, for each admissible tuple $(K_1(\cdot), K_2(\cdot))$, the path-wise performance measure (6), which contains trade-offs between dynamic agent coordination and system performance, is now rewritten as

$$J(K_1(\cdot), K_2(\cdot)) = z^T(T)\hat{N}_T z(T) + \int_0^T z^T(t)\hat{N}(t)z(t)\mathrm{d}t \tag{23}$$

where the positive semi-definite terminal penalty $\hat{N}_T \in \mathbb{R}^{3n\times 3n}$ and positive definite transient weighting $\hat{N}(t) = \hat{N}(t, \omega) : [0, T] \times \Omega \mapsto \mathbb{R}^{3n\times 3n}$ are given by

$$\hat{N}_T \triangleq \begin{bmatrix} Q_T & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; \hat{N} \triangleq \begin{bmatrix} K_1^T R_1 K_1 + K_2^T R_2 K_2 + Q & -K_1^T R_1 K_1 & -K_2^T R_2 K_2 \\ -K_1^T R_1 K_1 & K_1^T R_1 K_1 & 0 \\ -K_2^T R_2 K_2 & 0 & K_2^T R_2 K_2 \end{bmatrix} \tag{24}$$

So far there are two types of information, that is, process information (19)–(21) and goal information (23)–(24) have been given in advance to cooperative agents 1 and 2. Because there is random disturbance of the process $\hat{w}(\cdot)$ affecting the overall

performance, both cooperative agents now need additional information about performance variations. This is *coupling information* and thus also known as *performance information*. It is natural to further assume that cooperative agents are risk averse. They both prefer to avoid the risks associated with collective performance. And, for the reason of measure of effectiveness, much of the discussion that follows will be concerned with the situation where cooperative agents have risk-averse attitudes toward all process random realizations.

Regarding the linear-quadratic structural constraints (19) and (23), the path-wise performance measure (23) with which the cooperative agents are coordinating their actions is clearly a random variable of the generalized chi-squared type. Hence, the degree of uncertainty of the path-wise performance measure (23) must be assessed via a complete set of higher-order statistics beyond the statistical mean or average. The essence of information about these higher-order performance-measure statistics in an attempt to describe or model performance uncertainty is now considered as a source of information flow, which will affect perception of the problem and the environment at each cooperative agent. Next, the question of how to characterize and influence performance information is answered by modeling and management of cumulants (also known as semi-invariants) associated with (23) as shown in the following result.

Theorem 1 *Let $z(\cdot)$ be a state variable of the stochastic cooperative dynamics (19) with initial values $z(\tau) \equiv z_\tau$ and $\tau \in [0, T]$. Further let the moment-generating function be denoted by*

$$\varphi(\tau, z_\tau, \theta) = \varrho(\tau, \theta) \exp\left\{z_\tau^T \Upsilon(\tau, \theta) z_\tau\right\} \tag{25}$$

$$\upsilon(\tau, \theta) = \ln\{\varrho(\tau, \theta)\}, \qquad \theta \in \mathbb{R}^+. \tag{26}$$

Then the cumulant-generating function has the form of quadratic affine

$$\psi(\tau, z_\tau, \theta) = z_\tau^T \Upsilon(\tau, \theta) z_\tau + \upsilon(\tau, \theta) \tag{27}$$

where the scalar solution $\upsilon(\tau, \theta)$ solves the backward-in-time differential equation

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\upsilon(\tau, \theta) = -\mathrm{Tr}\left\{\Upsilon(\tau, \theta)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\right\}, \qquad \upsilon(T, \theta) = 0 \tag{28}$$

and the matrix solution $\Upsilon(\tau, \theta)$ satisfies the backward-in-time differential equation

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}\Upsilon(\tau, \theta) = &-\hat{F}^T(\tau)\Upsilon(\tau, \theta) - \Upsilon(\tau, \theta)\hat{F}(\tau) \\ &- 2\Upsilon(\tau, \theta)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\Upsilon(\tau, \theta) - \theta\hat{N}(\tau), \qquad \Upsilon(T, \theta) = \theta\hat{N}^f.\end{aligned} \tag{29}$$

Meanwhile, the scalar solution $\varrho(\tau)$ satisfies the backward-in-time differential equation

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\varrho\left(\tau,\theta\right)=-\varrho\left(\tau,\theta\right)\mathrm{Tr}\left\{\Upsilon(\tau,\theta)\hat{G}\left(\tau\right)\hat{W}\hat{G}^{T}\left(\tau\right)\right\},\qquad\varrho\left(T,\theta\right)=1\,. \tag{30}$$

Proof. For notional simplicity, it is convenient to have $\varpi\left(\tau,z_{\tau},\theta\right)\triangleq\exp\left\{\theta J\left(\tau,z_{\tau}\right)\right\}$ in which the performance measure (23) is rewritten as the cost-to-go function from an arbitrary state z_{τ} at a running time $\tau\in[0,T]$, that is,

$$J(\tau,z_{\tau})=z^{T}(T)\hat{N}_{T}z(T)+\int_{\tau}^{T}z^{T}(t)\hat{N}(t)z(t)\mathrm{d}t \tag{31}$$

subject to

$$\mathrm{d}z(t)=\hat{F}(t)z(t)\mathrm{d}t+\hat{G}(t)\mathrm{d}\hat{w}(t)\,,\qquad z(\tau)=z_{\tau}\,. \tag{32}$$

By definition, the moment-generating function is $\varphi(\tau,z_{\tau},\theta)\triangleq E\left\{\varpi\left(\tau,z_{\tau},\theta\right)\right\}$. Thus, the total time derivative of $\varphi(\tau,z_{\tau},\theta)$ is obtained as

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\varphi\left(\tau,z_{\tau},\theta\right)=-\varphi\left(\tau,z_{\tau},\theta\right)\theta z_{\tau}^{T}\hat{N}(\tau)z_{\tau}\,.$$

Using the standard Ito's formula, it follows

$$\begin{aligned}
&\mathrm{d}\varphi\left(\tau,z_{\tau},\theta\right)=E\left\{\mathrm{d}\varpi\left(\tau,z_{\tau},\theta\right)\right\}\\
&=E\Big\{\varpi_{\tau}\left(\tau,z_{\tau},\theta\right)\mathrm{d}\tau+\varpi_{z_{\tau}}\left(\tau,z_{\tau},\theta\right)\mathrm{d}z_{\tau}+\frac{1}{2}\mathrm{Tr}\left\{\varpi_{z_{\tau}z_{\tau}}(\tau,z_{\tau},\theta)\hat{G}(\tau)\hat{W}\hat{G}^{T}(\tau)\right\}\mathrm{d}\tau\Big\},\\
&=\varphi_{\tau}(\tau,z_{\tau},\theta)\mathrm{d}\tau+\varphi_{z_{\tau}}(\tau,z_{\tau},\theta)\hat{F}(\tau)z_{\tau}\mathrm{d}\tau+\frac{1}{2}\mathrm{Tr}\left\{\varphi_{z_{\tau}z_{\tau}}(\tau,z_{\tau},\theta)\hat{G}(\tau)\hat{W}\hat{G}^{T}(\tau)\right\}\mathrm{d}\tau
\end{aligned}$$

which under the hypothesis of $\varphi\left(\tau,z_{\tau},\theta\right)=\varrho\left(\tau,\theta\right)\exp\left\{x_{\tau}^{T}\Upsilon_{a}(\tau,\theta)x_{\tau}\right\}$ and its partial derivatives

$$\begin{aligned}
\varphi_{\tau}\left(\tau,z_{\tau},\theta\right)&=\varphi\left(\tau,z_{\tau},\theta\right)\left[\frac{\frac{\mathrm{d}}{\mathrm{d}\tau}\varrho(\tau,\theta)}{\varrho(\tau,\theta)}+z_{\tau}^{T}\frac{\mathrm{d}}{\mathrm{d}\tau}\Upsilon(\tau,\theta)z_{\tau}\right]\\
\varphi_{z_{\tau}}\left(\tau,z_{\tau},\theta\right)&=\varphi\left(\tau,z_{\tau},\theta\right)z_{\tau}^{T}\left[\Upsilon(\tau,\theta)+\Upsilon^{T}(\tau,\theta)\right]\\
\varphi_{z_{\tau}z_{\tau}}\left(\tau,x_{\tau},\theta\right)&=\varphi\left(\tau,z_{\tau},\theta\right)\left[\Upsilon(\tau,\theta)+\Upsilon^{T}(\tau,\theta)\right]\\
&\quad+\varphi\left(\tau,z_{\tau},\theta\right)\left[\Upsilon(\tau,\theta)+\Upsilon^{T}(\tau,\theta)\right]z_{\tau}z_{\tau}^{T}\left[\Upsilon(\tau,\theta)+\Upsilon^{T}(\tau,\theta)\right]
\end{aligned}$$

leads to the result

$$
\begin{aligned}
-\varphi(\tau, z_\tau, \theta)\,\theta z_\tau^T \hat{N}(\tau) z_\tau = & \frac{\frac{\mathrm{d}}{\mathrm{d}\tau}\varrho(\tau,\theta)}{\varrho(\tau,\theta)}\varphi(\tau, z_\tau, \theta) + \varphi(\tau, z_\tau, \theta)\, z_\tau^T \frac{\mathrm{d}}{\mathrm{d}\tau}\Upsilon(\tau,\theta) z_\tau \\
& + \varphi(\tau, z_\tau, \theta)\, z_\tau^T \left[\hat{F}^T(\tau)\Upsilon(\tau,\theta) + \Upsilon(\tau,\theta)\hat{F}(\tau)\right] z_\tau \\
& + \varphi(\tau, z_\tau, \theta)\left[2 z_\tau^T \Upsilon(\tau,\theta)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\Upsilon(\tau,\theta) z_\tau\right. \\
& \left. + \mathrm{Tr}\left\{\Upsilon(\tau,\theta)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\right\}\right].
\end{aligned}
$$

To have constant and quadratic terms being independent of arbitrary z_τ, it requires

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\tau}\Upsilon(\tau,\theta) &= -\hat{F}^T(\tau)\Upsilon(\tau,\theta) - \Upsilon(\tau,\theta)\hat{F}(\tau) - 2\Upsilon(\tau,\theta)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\Upsilon(\tau,\theta) - \theta\hat{N}(\tau) \\
\frac{\mathrm{d}}{\mathrm{d}\tau}\varrho(\tau,\theta) &= -\varrho(\tau,\theta)\,\mathrm{Tr}\left\{\Upsilon(\tau,\theta)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\right\}
\end{aligned}
$$

with the terminal-value conditions $\Upsilon(T,\theta) = \theta\hat{N}_T$ and $\varrho(T,\theta) = 1$. Finally, the backward-in-time differential equation satisfied by $\upsilon(\tau,\theta)$ is obtained

$$
\frac{\mathrm{d}}{\mathrm{d}\tau}\upsilon(\tau,\theta) = -\mathrm{Tr}\left\{\Upsilon(\tau,\theta)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\right\}, \qquad \upsilon(T,\theta) = 0. \qquad \square
$$

As it turns out that all the higher-order characteristic distributions associated with performance uncertainty and risk are very well captured in the higher-order performance-measure statistics associated with (31). Subsequently, higher-order statistics that encapsulate the uncertain nature of (31) can now be generated via a MacLaurin series of the cumulant-generating function (27)

$$
\psi(\tau, z_\tau, \theta) \triangleq \sum_{r=1}^{\infty} \kappa_r(z_\tau)\frac{\theta^r}{r!} = \sum_{r=1}^{\infty} \frac{\partial^{(r)}}{\partial\theta^{(r)}}\psi(\tau, z_\tau, \theta)\bigg|_{\theta=0}\frac{\theta^r}{r!} \tag{33}
$$

in which $\kappa_r(z_\tau)$'s are called performance-measure statistics. Moreover, the series expansion coefficients are computed by using the cumulant-generating function (27)

$$
\frac{\partial^{(r)}}{\partial\theta^{(r)}}\psi(\tau, z_\tau, \theta)\bigg|_{\theta=0} = z_\tau^T \frac{\partial^{(r)}}{\partial\theta^{(r)}}\Upsilon(\tau,\theta)\bigg|_{\theta=0} z_\tau + \frac{\partial^{(r)}}{\partial\theta^{(r)}}\upsilon(\tau,\theta)\bigg|_{\theta=0}. \tag{34}
$$

In view of the definition (33), the rth performance-measure statistic therefore follows

$$
\kappa_r(z_\tau) = z_\tau^T \frac{\partial^{(r)}}{\partial\theta^{(r)}}\Upsilon(\tau,\theta)\bigg|_{\theta=0} z_\tau + \frac{\partial^{(r)}}{\partial\theta^{(r)}}\upsilon(\tau,\theta)\bigg|_{\theta=0} \tag{35}
$$

for any finite $1 \leq r < \infty$. For notational convenience, the following change of notations:

$$H_r(\tau) \triangleq \frac{\partial^{(r)}}{\partial\theta^{(r)}}\Upsilon(\tau,\theta)\bigg|_{\theta=0} \quad \text{and} \quad D_r(\tau) \triangleq \frac{\partial^{(r)}}{\partial\theta^{(r)}}\upsilon(\tau,\theta)\bigg|_{\theta=0} \tag{36}$$

are introduced so that the next theorem provides an effective and accurate capability for forecasting all the higher-order characteristics associated with performance uncertainty. Therefore, via higher-order performance-measure statistics and adaptive decision making, it is anticipated that future performance variations will lose the element of surprise due to the inherent property of self-enforcing and risk-averse decision solutions that are readily capable of reshaping the cumulative probability distribution of closed-loop performance.

Theorem 2 *Performance-Measure Statistics*

Let the stochastic two-agent cooperative system be described by (19) and (23) in which the pairs (A, B_1) and (A, B_2) are uniformly stabilizable and the pairs (A, C_1) and (A, C_2) are uniformly detectable. For $k \in \mathbb{N}$ fixed, the kth cumulant of performance measure (23) is given by

$$\kappa_k(z_0) = z_0^T H_k(0) z_0 + D_k(0) \tag{37}$$

where the supporting variables $\{H_r(\tau)\}_{r=1}^k$ and $\{D_r(\tau)\}_{r=1}^k$ evaluated at $\tau = 0$ satisfy the differential equations (with the dependence of $H_r(\tau)$ and $D_r(\tau)$ upon $K_1(\tau)$, $K_2(\tau)$, $L_1(\tau)$ and $L_2(\tau)$ suppressed)

$$\frac{\mathrm{d}}{\mathrm{d}\tau}H_1(\tau) = -\hat{F}^T(\tau)H_1(\tau) - H_1(\tau)\hat{F}(\tau) - \hat{N}(\tau) \tag{38}$$

$$\frac{\mathrm{d}}{\mathrm{d}\tau}H_r(\tau) = -\hat{F}^T(\tau)H_r(\tau) - H_r(\tau)\hat{F}(\tau) - \sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}H_s(\tau)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)H_{r-s}(\tau), \qquad 2 \leq r \leq k \tag{39}$$

$$\frac{\mathrm{d}}{\mathrm{d}\tau}D_r(\tau) = -\mathrm{Tr}\left\{H_r(\tau)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\right\}, \qquad 1 \leq r \leq k \tag{40}$$

where the terminal-value conditions $H_1(T) = \hat{N}_T$, $H_r(T) = 0$ for $2 \leq r \leq k$ and $D_r(T) = 0$ for $1 \leq r \leq k$.

Proof. The expression of performance-measure statistics described in (37) is readily justified by using result (35) and definition (36). What remains is to show that the solutions $H_r(\tau)$ and $D_r(\tau)$ for $1 \leq r \leq k$ indeed satisfy the dynamical equations (38)–(40). Notice that the dynamical equations (38)–(40) are satisfied by the solutions $H_r(\tau)$ and $D_r(\tau)$ and can be obtained by successively taking time

derivatives with respect to θ of the supporting equations (28)–(29) together with the assumption of (A, B_1) and (A, B_2) being uniformly stabilizable on $[0, T]$. □

5 Cooperative Decision Strategies with Risk Aversion

The purpose of this section is to provide statements of the optimal statistical control with the addition of the necessary and sufficient conditions for optimality for the stochastic two-agent cooperative control and decision problem that are considered in this research investigation. The optimal statistical control of stochastic two-agent cooperative systems herein is distinguished by the fact that the evolution in time of all mathematical statistics (37) associated with the random performance measure (23) of the generalized chi-squared type are naturally described by means of matrix differential equations (38)–(40).

For such problems it is important to have a compact statement of the optimal statistical control so as to aid mathematical manipulation. To make this more precise, one may think of the k-tuple state variables $\mathcal{H}(\cdot) \triangleq (\mathcal{H}_1(\cdot), \ldots, \mathcal{H}_k(\cdot))$ and $\mathcal{D}(\cdot) \triangleq (\mathcal{D}_1(\cdot), \ldots, \mathcal{D}_k(\cdot))$ whose continuously differentiable states $\mathcal{H}_r \in \mathcal{C}^1([0, T]; \mathbb{R}^{3n\times 3n})$ and $\mathcal{D}_r \in \mathcal{C}^1([0, T]; \mathbb{R})$ having the representations $\mathcal{H}_r(\cdot) \triangleq H_r(\cdot)$ and $\mathcal{D}_r(\cdot) \triangleq D_r(\cdot)$ with the right members satisfying the dynamics (38)–(40) are defined on $[0, T]$. In the remainder of the development, the convenient mappings are introduced as follows

$$\mathcal{F}_r : [0, T] \times (\mathbb{R}^{3n\times 3n})^k \mapsto \mathbb{R}^{3n\times 3n}$$
$$\mathcal{G}_r : [0, T] \times (\mathbb{R}^{3n\times 3n})^k \mapsto \mathbb{R}$$

where the rules of action are given by

$$\mathcal{F}_1(\tau, \mathcal{H}) \triangleq -\hat{F}^T(\tau)\mathcal{H}_1(\tau) - \mathcal{H}_1(\tau)\hat{F}(\tau) - \hat{N}(\tau)$$
$$\mathcal{F}_r(\tau, \mathcal{H}) \triangleq -\hat{F}^T(\tau)\mathcal{H}_r(\tau) - \mathcal{H}_r(\tau)\hat{F}(\tau)$$
$$- \sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!}\mathcal{H}_s(\tau)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\mathcal{H}_{r-s}(\tau), \qquad 2 \le r \le k$$
$$\mathcal{G}_r(\tau, \mathcal{H}) \triangleq -\mathrm{Tr}\left\{\mathcal{H}_r(\tau)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\right\}, \qquad \le 1 \le r \le k.$$

The product mappings that follow are necessary for a compact formulation

$$\mathcal{F}_1 \times \cdots \times \mathcal{F}_k : [0, T] \times (\mathbb{R}^{3n\times 3n})^k \mapsto (\mathbb{R}^{3n\times 3n})^k$$
$$\mathcal{G}_1 \times \cdots \times \mathcal{G}_k : [0, T] \times (\mathbb{R}^{3n\times 3n})^k \mapsto \mathbb{R}^k$$

whereby the corresponding notations $\mathcal{F} \triangleq \mathcal{F}_1 \times \cdots \times \mathcal{F}_k$ and $\mathcal{G} \triangleq \mathcal{G}_1 \times \cdots \times \mathcal{G}_k$ are used. Thus, the dynamic equations of motion (38)–(40) can be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathcal{H}(\tau) = \mathcal{F}(\tau, \mathcal{H}(\tau)), \qquad \mathcal{H}(T) \equiv \mathcal{H}_T \tag{41}$$

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathcal{D}(\tau) = \mathcal{G}\left(\tau, \mathcal{H}(\tau)\right), \qquad \mathcal{D}(T) \equiv \mathcal{D}_T \tag{42}$$

where k-tuple values $\mathcal{H}_T \triangleq \left(\hat{N}_T, 0, \ldots, 0\right)$ and $\mathcal{D}_T = (0, \ldots, 0)$.

Notice that the product system uniquely determines the state matrices $\mathcal{H}$ and $\mathcal{D}$ once the admissible feedback gain K_1 and K_2 together with admissible filtering gains L_1 and L_2 being specified. Henceforth, these state variables will be considered as $\mathcal{H} \equiv \mathcal{H}(\cdot, K_1, K_2, L_1, L_2)$ and $\mathcal{D} \equiv \mathcal{D}(\cdot, K_1, K_2, L_1, L_2)$. The performance index in optimal statistical control problems can now be formulated in K_1, K_2, L_1, and L_2. For the given terminal data $(T, \mathcal{H}_T, \mathcal{D}_T)$, the classes of admissible feedback and filter gains are next defined.

Definition 1 *Admissible Filter and Feedback Gains*

Let compact subsets $\overline{L}_i \subset \mathbb{R}^{n\times r_i}$ and $\overline{K}_i \subset \mathbb{R}^{m_i\times n}$ and $i = 1, 2$ be the sets of allowable filter and feedback gain values. For the given $k \in \mathbb{N}$ and sequence $\mu = \{\mu_r \geq 0\}_{r=1}^k$ with $\mu_1 > 0$, the set of admissible filter gains $\mathcal{L}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}$ and feedback gains $\mathcal{K}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}$ are, respectively, assumed to be the classes of $\mathcal{C}([0, T]; \mathbb{R}^{n\times r_i})$ and $\mathcal{C}([0, T]; \mathbb{R}^{m_i\times n})$ with values $L_i(\cdot) \in \overline{L}_i$ and $K_i(\cdot) \in \overline{K}_i$ for which solutions to the dynamic equations (41)–(42) with the terminal-value conditions $\mathcal{H}(T) = \mathcal{H}_T$ and $\mathcal{D}(T) = \mathcal{D}_T$ exist on the interval of optimization $[0, T]$.

It is now crucial to plan for robust decisions and performance reliability from the start because it is going to be much more difficult and expensive to add reliability to the process later. To be used in the design process, performance-based reliability requirements must be verifiable by analysis; in particular, they must be measurable, like all higher-order performance-measure statistics, as evidenced in the previous section. These higher-order performance-measure statistics become the test criteria for the requirement of performance-based reliability. What follows is risk-value aware performance index in the optimal statistical control. It naturally contains some trade-offs between performance values and risks for the subject class of stochastic decision problems.

On the Cartesian product $\mathcal{L}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{K}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}$ and $i = 1, 2$, the performance index with risk-value considerations in the optimal statistical control is subsequently defined as follows.

Definition 2 *Risk-Value Aware Performance Index*

Fix $k \in \mathbb{N}$ and the sequence of scalar coefficients $\mu = \{\mu_r \geq 0\}_{r=1}^k$ with $\mu_1 > 0$. Then for the given z_0, the risk-value aware performance index

$$\phi_0 : (\mathbb{R}^{3n\times 3n})^k \times \mathbb{R}^k \mapsto \mathbb{R}^+$$

pertaining to the optimal statistical control of the stochastic cooperative decision problem involved two agents over $[0, T]$ *is defined by*

$$\phi_0(\mathcal{H}, \mathcal{D}) \triangleq \underbrace{\mu_1 \kappa_1(z_0)}_{\text{Value Measure}} + \underbrace{\mu_2 \kappa_2(z_0) + \cdots + \mu_k \kappa_k(z_0)}_{\text{Risk Measures}}$$
$$= \sum_{r=1}^{k} \mu_r \left[z_0^T \mathcal{H}_r(0) z_0 + \mathcal{D}_r(0) \right] \tag{43}$$

where additional design freedom by means of μ_r*'s utilized by cooperative agents are sufficient to meet and exceed different levels of performance-based reliability requirements, for instance, mean (i.e., the average of performance measure), variance (i.e., the dispersion of values of performance measure around its mean), skewness (i.e., the antisymmetry of the density of performance measure), kurtosis (i.e., the heaviness in the density tails of performance measure), etc., pertaining to closed-loop performance variations and uncertainties while the cumulant-generating solutions* $\{\mathcal{H}_r(\tau)\}_{r=1}^k$ *and* $\{\mathcal{D}_r(\tau)\}_{r=1}^k$ *evaluated at* $\tau = 0$ *satisfy the dynamical equations (41)–(42).*

Notice that the assumption of all $\mu_r \geq 0$ with $\mu_1 > 0$ and $r = 1, \ldots, k$ in the definition of risk-averse performance index is assumed for strictly order preserving and well-posedness of the optimization at hand. This assumption, however, cannot be always justified, because human subjects are also well known to exhibit risk-taking patterns in certain situations (e.g., when higher values of dispersion are preferred).

Next, the optimization statement for the statistical control of the stochastic cooperative system for two agents over a finite horizon is stated.

Definition 3 *Optimization Problem*

Given $\mu_1, \ldots, \mu_k$ *with* $\mu_1 > 0$*, the optimization problem of the statistical control over* $[0, T]$ *is given by*

$$\min_{L_i(\cdot) \in \mathcal{L}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}, K_i(\cdot) \in \mathcal{K}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}} \phi_0(\mathcal{H}, \mathcal{D}), \qquad i = 1, 2 \tag{44}$$

subject to the dynamical equations (41)–(42) for $\tau \in [0, T]$.

Opposite to the spirit of the earlier work by the authors [5, 6] relative to the traditional approach of dynamic programming to the optimization problem of Mayer form, the problem (44) of finding extremals may, however, be recast as that of minimizing the fixed-time optimization problem in Bolza form, that is,

$$\phi_0(0, \mathcal{X}) = \mathrm{Tr}\left\{ \mathcal{X}(0) z_0 z_0^T \right\} + \int_0^T \mathrm{Tr}\left\{ \mathcal{X}(t) \hat{G}(t) \hat{W} \hat{G}^T(t) \right\} \mathrm{d}t \tag{45}$$

subject to

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}\mathcal{X}(\tau) =& -\hat{F}^T(\tau)\mathcal{X}(\tau) - \mathcal{X}(\tau)\hat{F}(\tau) - \mu_1\hat{N}(\tau)\\ &- \sum_{r=2}^{k}\mu_r\sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}\mathcal{H}_s(\tau)\hat{G}(\tau)\hat{W}\hat{G}^T(\tau)\mathcal{H}_{r-s}(\tau)\,,\\ \mathcal{M}(T) =& \mu_1\hat{N}_T\end{aligned} \tag{46}$$

wherein $\mathcal{X}(\tau) \triangleq \mu_1\mathcal{H}_1(\tau) + \cdots + \mu_k\mathcal{H}_k(\tau)$ and $\{\mathcal{H}_r(\tau)\}_{r=1}^k$ are satisfying the dynamical equations (38)–(40) for all $\tau \in [0, T]$.

Furthermore, the transformation of problem (45) and (46) into the framework required by the matrix minimum principle [7] that makes it possible to apply Pontryagin's results directly to problems whose state variables are most conveniently regarded as matrices is complete if further changes of variables are introduced, that is, $T-t = \tau$ and $\mathcal{X}(T-t) = \mathcal{M}(t)$. Thus, the aggregate equation (46) is rewritten as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{M}(t) =& \hat{F}^T(t)\mathcal{M}(t) + \mathcal{M}(t)\hat{F}(t) + \mu_1\hat{N}(t)\\ &+ \sum_{r=2}^{k}\mu_r\sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}\mathcal{H}_s(t)\hat{G}(t)\hat{W}\hat{G}^T(t)\mathcal{H}_{r-s}(t), \quad \mathcal{M}(0) = \mu_1\hat{N}_T.\end{aligned} \tag{47}$$

Now the matrix coefficients $\hat{F}$, $\hat{N}$, and $\hat{G}\hat{W}\hat{G}^T$ of the composite dynamics (19) for agent interaction and estimation are next partitioned to conform with the n-dimensional structure of (3) by means of

$$I_0^T \triangleq \begin{bmatrix} I & 0 & 0 \end{bmatrix}, \qquad I_1^T \triangleq \begin{bmatrix} 0 & I & 0 \end{bmatrix}, \qquad I_2^T \triangleq \begin{bmatrix} 0 & 0 & I \end{bmatrix}$$

where I is an $n \times n$ identity matrix and

$$\begin{aligned}\hat{F} =& I_0AI_0^T + I_1AI_1^T + I_2AI_2^T + I_0B_1K_1(I_0 - I_1)^T + I_2B_1K_1(I_2 - I_1)^T\\ &+ I_0B_2K_2(I_0 - I_2)^T + I_1B_2K_2(I_1 - I_2)^T - I_1L_1C_1I_1^T - I_2L_2C_2I_2^T\end{aligned} \tag{48}$$

$$\hat{N} = (I_0 - I_1)K_1^TR_1K_1(I_0 - I_1)^T + (I_0 - I_2)K_2^TR_2K_2(I_0 - I_2)^T + I_0QI_0^T \tag{49}$$

$$\begin{aligned}\hat{G}\hat{W}\hat{G}^T =& I_0GWG^TI_1^T + I_1GWG^TI_0^T + I_1GWG^TI_1^T + (I_0 + I_2)GWG^T(I_0 + I_2)^T\\ &+ I_1L_1V_1L_1^TI_1^T + I_2L_2V_2L_2^TI_2^T\,.\end{aligned} \tag{50}$$

Assume that $\mathcal{L}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{K}^i_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}$ and $i = 1, 2$ are nonempty and convex in $\mathbb{R}^{n\times r_i} \times \mathbb{R}^{m_i\times n}$. For all $(t, K_1, K_2, L_1, L_2) \in [0, T] \times \mathcal{K}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{K}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{L}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{L}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}$, the maps $h(\mathcal{M})$ and $q(t, \mathcal{M}(t, K_1, K_2, L_1, L_2))$ having the property of twice continuously differentiable, as defined from the risk-value aware performance index (45)

$$\begin{aligned}
&\phi_0(L_1(\cdot), K_1(\cdot), L_2(\cdot), K_2(\cdot)) \\
&\quad = h(\mathcal{M}(T)) + \int_0^T q(t, \mathcal{M}(t, K_1(t), K_2(t), L_1(t), L_2(t)))\mathrm{d}t \\
&\quad = \mathrm{Tr}\left\{\mathcal{M}(T)z_0 z_0^T\right\} + \int_0^T \mathrm{Tr}\left\{\mathcal{M}(t)\hat{G}(t)\hat{W}\hat{G}^T(t)\right\}\mathrm{d}t
\end{aligned} \tag{51}$$

are supposed to have all partial derivatives with respect to $\mathcal{M}$ up to order 2 being continuous in $(\mathcal{M}, K_1, K_2, L_1, L_2)$ with appropriate growths.

Moreover, any 4-tuple $(K_1^*, K_2^*, L_1^*, L_2^*) \in \mathcal{K}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{K}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{L}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{L}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}$ minimizing the risk-value aware performance index (51) is called optimal strategies with risk aversion of the optimization problem (44). The corresponding state process $\mathcal{M}^*(\cdot)$ is called an optimal state process. Further denote $\mathcal{P}(t)$ by the costate matrix associated with $\mathcal{M}(t)$ for each $t \in [0, T]$. The scalar Hamiltonian function for the optimization problem (47) and (51) is thus defined by

$$\begin{aligned}
\mathcal{V}(t, \mathcal{M}, K_1, K_2, L_1, L_2) \triangleq &\mathrm{Tr}\left\{\mathcal{M}\hat{G}\hat{W}\hat{G}^T\right\} + \mathrm{Tr}\Bigg\{\Bigg[\hat{F}^T\mathcal{M} + \mathcal{M}\hat{F} + \mu_1\hat{N} \\
&+ \sum_{r=2}^{k}\mu_r\sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}\mathcal{H}_s\hat{G}\hat{W}\hat{G}^T\mathcal{H}_{r-s}\Bigg]\mathcal{P}^T(t)\Bigg\}.
\end{aligned} \tag{52}$$

whereby in view of (48)–(50), the matrix variables $\mathcal{M}$, $\hat{F}$, $\hat{N}$, etc. shall be considered as $\mathcal{M}(t, K_1, K_2, L_1, L_2)$, $\hat{F}(t, K_1, K_2, L_1, L_2)$, $\hat{N}(t, K_1, K_2)$, etc., respectively.

Using the matrix minimum principle [7], the set of first-order necessary conditions for K_1^*, K_2^*, L_1^*, and L_2^* to be extremizers is composed of

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{M}^*(t) = \left.\frac{\partial\mathcal{V}}{\partial\mathcal{P}}\right|_* &= (\hat{F}^*)^T(t)\mathcal{M}^*(t) + \mathcal{M}^*(t)\hat{F}^*(t) + \mu_1\hat{N}^*(t) \\
&+ \sum_{r=2}^{k}\mu_r\sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}\mathcal{H}_s^*(t)\hat{G}^*(t)\hat{W}(\hat{G}^*)^T(t)\mathcal{H}_{r-s}^*(t), \quad \mathcal{M}^*(0) = \mu_1\hat{N}_T
\end{aligned} \tag{53}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{P}^*(t) = -\left.\frac{\partial \mathcal{V}}{\partial \mathcal{M}}\right|_* = -\hat{F}^*(t)\mathcal{P}^*(t) - \mathcal{P}^*(t)(\hat{F}^*)^T(t) - \hat{G}^*(t)\hat{W}(\hat{G}^*)^T(t)$$
$$\mathcal{P}^*(T) = z_0 z_0^T . \tag{54}$$

In addition, if $(K_1^*, K_2^*, L_1^*, L_2^*)$ is a local extremum of (52), it implies that

$$\mathcal{V}(t, \mathcal{M}^*(t), K_1, K_2, L_1, L_2) - \mathcal{V}(t, \mathcal{M}^*(t), K_1^*(t), K_2^*(t), L_1^*(t), L_2^*(t)) \geq 0 \tag{55}$$

for all $(K_1, K_2, L_1, L_2) \in \mathcal{K}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{K}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{L}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu} \times \mathcal{L}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}$ and $t \in [0, T]$. That is,

$$\min_{(K_1,K_2,L_1,L_2)\in \mathcal{K}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}\times\mathcal{K}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}\times\mathcal{L}^1_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}\times\mathcal{L}^2_{T,\mathcal{H}_T,\mathcal{D}_T;\mu}} \mathcal{V}(t, \mathcal{M}^*(t), K_1, K_2, L_1, L_2)$$
$$= \mathcal{V}(t, \mathcal{M}^*(t), K_1^*(t), K_2^*(t), L_1^*(t), L_2^*(t)) = 0 , \qquad \forall\, t \in [0, T] . \tag{56}$$

Equivalently, it follows that

$$0 \equiv \left.\frac{\partial \mathcal{V}}{\partial K_1}\right|_* = 2B_1^T(t)\Big[I_0^T \mathcal{M}^*(t)\mathcal{P}^*(t)(I_0 - I_1) + I_2^T \mathcal{M}^*(t)\mathcal{P}^*(t)(I_2 - I_1)\Big]$$
$$+ 2\mu_1 R_1(t)K_1(I_0 - I_1)^T\mathcal{P}^*(t)(I_0 - I_1) \tag{57}$$
$$0 \equiv \left.\frac{\partial \mathcal{V}}{\partial K_2}\right|_* = 2B_2^T(t)\Big[I_0^T \mathcal{M}^*(t)\mathcal{P}^*(t)(I_0 - I_2) + I_1^T \mathcal{M}^*(t)\mathcal{P}^*(t)(I_1 - I_2)\Big]$$
$$+ 2\mu_1 R_2(t)K_2(I_0 - I_2)^T\mathcal{P}^*(t)(I_0 - I_2) \tag{58}$$
$$0 \equiv \left.\frac{\partial \mathcal{V}}{\partial L_1}\right|_* = -2I_1^T \mathcal{M}^*(t)\mathcal{P}^*(t)I_1 C_1^T(t) + 2I_1^T \mathcal{M}^*(t)I_1 L_1 V_1$$
$$+ 2\sum_{r=2}^{k}\mu_2 \sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!} I_1^T \mathcal{H}_s^*(t)\mathcal{P}^*(t)\mathcal{H}_{r-s}^*(t)I_1 L_1 V_1 \tag{59}$$
$$0 \equiv \left.\frac{\partial \mathcal{V}}{\partial L_2}\right|_* = -2I_2^T \mathcal{M}^*(t)\mathcal{P}^*(t)I_2 C_2^T(t) + 2I_2^T \mathcal{M}^*(t)I_2 L_2 V_2$$
$$+ 2\sum_{r=2}^{k}\mu_2 \sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!} I_2^T \mathcal{H}_s^*(t)\mathcal{P}^*(t)\mathcal{H}_{r-s}^*(t)I_2 L_2 V_2 . \tag{60}$$

Furthermore, the second-order sufficient conditions that ensure the Hamiltonian functional (52) achieving its local minimum, require the following Hessian matrices to be positive definite; in particular,

$$\left.\frac{\partial^2 \mathcal{V}}{\partial K_1^2}\right|_* = 2\mu_1 R_1(t) \otimes (I_0 - I_1)^T \mathcal{P}^*(t)(I_0 - I_1) \tag{61}$$

$$\left.\frac{\partial^2 \mathcal{V}}{\partial K_2^2}\right|_* = 2\mu_1 R_2(t) \otimes (I_0 - I_2)^T \mathcal{P}^*(t)(I_0 - I_2) \tag{62}$$

$$\left.\frac{\partial^2 \mathcal{V}}{\partial L_1^2}\right|_* = 2I_1^T \left[\mathcal{M}^*(t) + \sum_{r=2}^{k} \mu_r \sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!} \mathcal{H}_s^*(t)\mathcal{P}^*(t)\mathcal{H}_{r-s}^*(t)\right] I_1 \otimes V_1 \tag{63}$$

$$\left.\frac{\partial^2 \mathcal{V}}{\partial L_2^2}\right|_* = 2I_2^T \left[\mathcal{M}^*(t) + \sum_{r=2}^{k} \mu_r \sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!} \mathcal{H}_s^*(t)\mathcal{P}^*(t)\mathcal{H}_{r-s}^*(t)\right] I_2 \otimes V_2 \tag{64}$$

wherein $\otimes$ stands for the Kronecker matrix product operator.

By the matrix variation of constants formula [8], the matrix solutions of the cumulant-generating Eqs. (38)–(39) and the costate Eq. (54) can be rewritten in the integral forms, for each $\tau \in [0, T]$

$$\mathcal{H}_1^*(\tau) = \Phi^T(T,\tau)\hat{N}_T\Phi(T,\tau) + \int_\tau^T \Phi(T,t)\hat{N}^*(t)\Phi(T,t)\mathrm{d}t \tag{65}$$

$$\mathcal{H}_r^*(\tau) = \int_\tau^T \Phi(T,t) \sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!} \mathcal{H}_s^*(t)\hat{G}^*(t)\hat{W}(\hat{G}^*)^T(t)\mathcal{H}_{r-s}^*(t)\Phi(T,t)\mathrm{d}t \tag{66}$$

$$\mathcal{P}^*(\tau) = \Phi^T(T,\tau)z_0 z_0^T \Phi(T,\tau) + \int_\tau^T \Phi(T,t)\hat{G}^*(t)\hat{W}(\hat{G}^*)^T(t)\Phi(T,t)\mathrm{d}t \tag{67}$$

provided that

$$\frac{\mathrm{d}}{\mathrm{d}t}\Phi(t,0) = \hat{F}^*(t)\Phi(t,0), \qquad \Phi(0,0) = I. \tag{68}$$

It can easily be verified that the following matrix inequalities hold for all $t \in [0, T]$

$$\begin{aligned}
\hat{N}_T &\geq 0 \\
\hat{N}^*(\cdot) &> 0 \\
\sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!} \mathcal{H}_s^*(\cdot)\hat{G}^*(\cdot)\hat{W}(\hat{G}^*)^T(\cdot)\mathcal{H}_{r-s}^*(\cdot) &\geq 0 \\
z_0 z_0^T &\geq 0 \\
\hat{G}^*(\cdot)\hat{W}(\hat{G}^*)^T(\cdot) &> 0\,.
\end{aligned}$$

Therefore, it implies that $\{\mathcal{H}_r^*(\cdot)\}_{r=1}^k$ and thus $\mathcal{M}^*(\cdot)$, as well as $\mathcal{P}^*(\cdot)$ with the integral forms (65)–(67), are positive definite on $[0, T]$. Subsequently, one can show that the following matrix inequalities are valid

$$(I_0 - I_1)^T \mathcal{P}^*(\cdot)(I_0 - I_1) > 0 \tag{69}$$

$$(I_0 - I_2)^T \mathcal{P}^*(\cdot)(I_0 - I_2) > 0 \tag{70}$$

$$I_1^T \left[\mathcal{M}^*(\cdot) + \sum_{r=2}^{k} \mu_r \sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!} \mathcal{H}_s^*(\cdot) \mathcal{P}^*(\cdot) \mathcal{H}_{r-s}^*(\cdot) \right] I_1 > 0 \tag{71}$$

$$I_2^T \left[\mathcal{M}^*(\cdot) + \sum_{r=2}^{k} \mu_r \sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!} \mathcal{H}_s^*(\cdot) \mathcal{P}^*(\cdot) \mathcal{H}_{r-s}^*(\cdot) \right] I_2 > 0\,. \tag{72}$$

In view of (69)–(72), all the Hessian matrices (61)–(64) are thus positive definite. As the result, the local extremizer $(K_1^*, K_2^*, L_1^*, L_2^*)$ formed by the first-order necessary conditions (57)–(60) becomes a local minimizer.

Notice that the results (53)–(60) are coupled forward-in-time and backward-in-time matrix-valued differential equations. Putting the corresponding state and costate equations together, the following optimality system for cooperative decision strategies with risk aversion are summarized as follows.

Theorem 3 *Let (A, B_i) and (A, C_i) for $i = 1, 2$ be uniformly stabilizable and detectable. Suppose that $u_i(\cdot) = K_i(\cdot)\hat{x}_i(\cdot) \in U_i$; the common state and local measurement processes are defined by (3)–(5); and the decentralized filters with $L_i(\cdot)$ are governed by (7). Then cooperative decision and control strategies $u_1(\cdot)$ and $u_2(\cdot)$ with risk aversion supported by the optimal pairs $(K_1^*(\cdot), L_1^*(\cdot))$ and $(K_2^*(\cdot), L_2^*(\cdot))$ are given by*

$$K_1^*(t) = -R_1^{-1}(t) B_1^T(t) \Bigg[I_0^T \sum_{r=1}^{k} \mu_r \mathcal{H}_r^*(t)\, \mathcal{P}^*(T-t)(I_0 - I_1) + I_2^T \sum_{r=1}^{k} \mu_r \mathcal{H}_r^*(t) \cdot \mathcal{P}^*(T-t)(I_2 - I_1) \Bigg] \left[(I_0 - I_1)^T \mathcal{P}^*(T-t)(I_0 - I_1) \right]^{-1} \tag{73}$$

$$L_1^*(t) = \left\{ I_1^T \left[\sum_{r=1}^{k} \mu_r \mathcal{H}_r^*(t) + \sum_{r=2}^{k} \mu_r \sum_{s=1}^{r-1} \frac{2r!}{s!(r-s)!} \mathcal{H}_s^*(t) \mathcal{P}^*(T-t) \mathcal{H}_{r-s}^*(t) \right] I_1 \right\}^{-1} \cdot I_1^T \sum_{r=1}^{k} \mu_r \mathcal{H}_r^*(t)\, \mathcal{P}^*(T-t) I_1 C_1^T(t) V_1^{-1} \tag{74}$$

and

$$K_2^*(t) = -R_2^{-1}(t)B_2^T(t)\Bigg[I_0^T\sum_{r=1}^{k}\mu_r\mathcal{H}_r^*(t)\,\mathcal{P}^*(T-t)(I_0-I_2)+I_1^T\sum_{r=1}^{k}\mu_r\mathcal{H}_r^*(t)$$
$$\cdot\,\mathcal{P}^*(T-t)(I_1-I_2)\Bigg]\Big[(I_0-I_2)^T\mathcal{P}^*(T-t)(I_0-I_2)\Big]^{-1} \tag{75}$$

$$L_2^*(t) = \Bigg\{I_2^T\Big[\sum_{r=1}^{k}\mu_r\mathcal{H}_r^*(t)+\sum_{r=2}^{k}\mu_r\sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}\mathcal{H}_s^*(t)\mathcal{P}^*(T-t)\mathcal{H}_{r-s}^*(t)\Big]I_2\Bigg\}^{-1}$$
$$\cdot\,I_2^T\sum_{r=1}^{k}\mu_r\mathcal{H}_r^*(t)\,\mathcal{P}^*(T-t)I_2C_2^T(t)V_2^{-1} \tag{76}$$

where the optimal state solutions $\{\mathcal{H}_r^*(\cdot)\}_{r=1}^k$ *supporting all the statistics for performance robustness and risk-averse decisions are governed by the forward-in-time matrix-valued differential equations with the terminal-value conditions* $\mathcal{H}_1^*(0)=\hat{N}_T$ *and* $\mathcal{H}_r^*(0)=0$ *for* $2\le r\le k$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}_1^*(t) = (\hat{F}^*)^T(t)\mathcal{H}_1^*(t)+\mathcal{H}_1^*(t)\hat{F}^*(t)+\hat{N}^*(t) \tag{77}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}_r^*(t) = (\hat{F}^*)^T(t)\mathcal{H}_r^*(t)+\mathcal{H}_r^*(t)\hat{F}^*(t)$$
$$+\sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}\mathcal{H}_s^*(t)\hat{G}^*(t)\hat{W}(\hat{G}^*)^T(t)\mathcal{H}_{r-s}^*(t) \tag{78}$$

and the optimal costate solution $\mathcal{P}^*(\cdot)$ *satisfies the backward-in-time matrix-valued differential equation with the terminal-value condition* $\mathcal{P}^*(T)=z_0z_0^T$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{P}^*(t) = -\hat{F}^*(t)\mathcal{P}^*(t)-\mathcal{P}^*(t)(\hat{F}^*)^T(t)-\hat{G}^*(t)\hat{W}(\hat{G}^*)^T(t)\,. \tag{79}$$

Remark 1. The results herein are certainly viewed as the generalization of those obtained from [9], where with respect to the subject of performance robustness, most developed work has fundamentally focused on the first-order assessment of performance variations through statistical averaging of performance measures of interest. To obtain the optimal values for cooperative control strategies, a two-point boundary value problem involving matrix differential equations must be solved. Moreover, the states $\{\mathcal{H}_r(\cdot)^*\}_{r=1}^k$ and costates $\mathcal{P}^*(\cdot)$ play an important role in the determination of cooperative decision strategies with risk aversion. Not only $K_i^*(\cdot)$ and $L_i^*(\cdot)$ for $i=1,2$ are tightly coupled. They also depend on the mathematical statistics associated with performance uncertainty; in particular,

mean, variance, skewness, etc. The need for the decision laws of cooperative strategies to take into account accurate estimations of performance uncertainty is one form of interaction between two interdependent functions of a decision strategy: (1) anticipation of performance uncertainty and (2) proactive decisions for mitigating performance riskiness. This form of interaction between these two decision strategy functions gives rise to what are now termed as *performance probing* and *performance cautioning* and thus are explicitly concerned in optimal statistical control of stochastic large-scale multi-agent systems [5, 6].

6 Conclusions

A new cooperative solution concept proposed herein is aimed at analytically addressing performance robustness, which is widely recognized as the pressing need in management control of stochastic multi-agent systems. One might consider typical applications in integrated situational awareness and socioeconomic problems in which the manager of an information-gathering department assigns his data-collecting group of people to perform such tasks as collect data, conduct polls, or research statistics so that an accurate forecast regarding future trends of the entire organization or agency can be carried out. In the most basic framework of performance-information analysis, a performance-information system transmits messages about higher-order characteristics of performance uncertainty to cooperative agents for use in future adaption of risk-averse decisions. The messages of performance-measure statistics transmitted are then influenced by the attributes of the interactive decision setting. Performance-measure statistics are now expected to work not only as feedback information for future risk-averse decisions but also as an influence mechanism for cooperative agents' behaviors. The solution of a matrix two-point boundary value problem will yield the optimal parameter values of cooperative decision strategies. Furthermore, the implementation of the analytical solution can be computationally intensive. Henceforth, the basic concept of successive approximation and thus a sequence of suboptimal control functions will be the emerging subject of future research investigation.

References

1. Pollatsek, A., Tversky, A.: Theory of risk. J. Math. Psychol. **7**, 540–53 (1970)
2. Luce, R.D.: Several possible measures of risk. Theory Decis. **12**, 217–228 (1980)
3. Radner, R.: Team decision problems. Ann. Math. Stat. **33**, 857–881 (1962)
4. Sandell, N.R. Jr., Varaiya, P., Athans, M., Safonov, M.G.: A survey of decentralized control methods for large-scale systems. IEEE Trans. Automat. Contr. **23**, 108–129 (1978)
5. Pham, K.D.: New results in stochastic cooperative games: strategic coordination for multi-resolution performance robustness. In: Hirsch, M.J., Pardalos, P.M., Murphey, R., Grundel, D. (eds.) Optimization and Cooperative Control Strategies. Series Lecture Notes in Control and Information Sciences, vol. 381, pp. 257–285 (2008)

6. Pham, K.D.: Performance-information analysis and distributed feedback stabilization in large-scale interconnected systems. In: Hirsch, M.J., Pardalos, P.M., Murphey, R. (eds.) Dynamics of Information Systems Theory and Applications Series: Springer Optimization and Its Applications, vol. 40, pp. 45–81. Springer, New York (2010). DOI:10.1007/978-1-4419-5689-7_3
7. Athans, M.: The matrix minimum prinicple. Inform. Contr. **11**, 592–606. Elsevier (1967)
8. Brockett, R.W.: Finite Dimensional Linear Systems. Wiley, New York (1970)
9. Chong, C.Y.: On the stochastic control of linear systems with different information sets. IEEE Trans. Automat. Contr. **16**(5), 423–430 (1971)

Modeling Interactions in Complex Systems: Self-Coordination, Game-Theoretic Design Protocols, and Performance Reliability-Aided Decision Making*

Khanh D. Pham and Meir Pachter

Abstract The subject of this research article is concerned with the development of approaches to modeling interactions in complex systems. A complex system contains a number of decision makers, who put themselves in the place of the other: to build a mutual model of other decision makers. Different decision makers have different influence in the sense that they will have control over—or at least be able to influence—different parts of the environment. Attention is first given to process models of operations among decision makers, for which the slow and fast-core design is based on a singularly perturbed model of complex systems. Next, self-coordination and Nash game-theoretic formulation are fundamental design protocols, lending themselves conveniently to modeling self-interest interactions, from which complete coalition among decision makers is not possible due to hierarchical macrostructure, information, or process barriers. Therefore, decision makers make decisions by assuming the others try to adversely affect their objectives and terms. Individuals will be expected to work in a decentralized manner. Finally, the standards and beliefs of the decision makers are threefold: (i) a high priority for performance-based reliability is made from the start through a means of performance-information analysis; (ii) a performance index has benefit and risk

*The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States (U.S.) Air Force, Department of Defense, or U.S. Government.

K.D. Pham
Air Force Research Laboratory, Space Vehicles Directorate, Kirtland Air Force Base, New Mexico 87117, USA
e-mail: AFRL.RVSV@kirtland.af.mil

M. Pachter (✉)
Air Force Institute of Technology, Department of Electrical and Computer Engineering, Wright-Patterson Air Force Base, Ohio 45433, USA
e-mail: meir.pachter@afit.edu

A. Sorokin et al. (eds.), *Dynamics of Information Systems: Mathematical Foundations*, Springer Proceedings in Mathematics & Statistics 20, DOI 10.1007/978-1-4614-3906-6_17,

awareness to ensure how much of the inherent or design-in reliability actually ends up in the developmental and operational phases; and (iii) risk-averse decision policies towards potential interference and noncooperation from the others.

Keywords Fast and slow interactions • Mutual modeling • Self-coordination • Performance-measure statistics • Risk-averse control decisions • Performance reliability • Minimax estimation • Stochastic Nash games • Dynamic programming

1 Introduction

Today, a new view of business operations, including sales, marketing, manufacturing, and design as inherently complex, computational, and adaptive systems, has been emerging. Complex systems are composed of intelligent adaptive decision makers constrained and enabled by their locations in networks linking decision makers and knowledge and by the tasks, in which they are engaged. Some of the techniques for dealing with the size and complexity of these complex systems are modularity, distribution, abstraction, and intelligence. Combining these techniques implies the use of intelligent, distributed modules—the concept of multi-model strategies for large-scale stochastic systems introduced in [1]. Therein, it was shown that in order to obtain near equilibrium Nash strategies, the decision makers need only to solve two decoupled low-order systems: a stochastic control problem in the fast time scale at local levels and a joint slow game problem with finite-dimensional state estimators. This is accomplished by leveraging the multi-model situation wherein each decision maker needs to model only his local dynamics and some aggregated dynamics of the rest of the system.

The intention of this research article is to extend the results [1] for two-person nonzero-sum Linear Quadratic Gaussian (LQG) Nash games, to robust decision making for multiperson quadratic decision problems toward performance values and risks. When measuring performance reliability, statistical analysis for probabilistic nature of performance uncertainty is relied on as part of the long-range assessment of reliability. One of the most widely used measures for performance reliability is the statistical mean or average to summarize the underlying performance variations. However, other aspects of performance distributions that do not appear in most of the existing progress are variance, skewness, and so forth. For instance, it may nevertheless be true that some performance with negative skewness appears riskier than performance with positive skewness when expectation and variance are held constant. If skewness does, indeed, play an essential role in determining the perception of risk, then the range of applicability of the present theory for stochastic control and operations research should be restricted, for example, to symmetric or equally skewed performance-measures.

Thus, for reliability reasons on performance distributions, the research investigation herein is unique when compared to the existing literature and results.

Specifically, technical merits and research contributions include an effective integration of performance-information analysis into risk-averse strategy selection for performance robustness and reliability requirements so that: (i) intrinsic performance variations caused by stationary environments are addressed concurrently with other performance requirements and (ii) trade-off analysis on performance benefits and risks directly evaluates the impact of reliability as well as other performance requirements. Hence, via higher-order performance-measure statistics and adaptive decision making, it is anticipated that future performance variations will lose the element of surprise due to the inherent property of self-enforcing and risk-averse decision solutions that are highly capable of reshaping probabilistic performance distributions at both macro- and microlevels.

The outline of this research begins with Sects. 2 and 3 that deal with subject of how to formulate complex systems with multiple time scales and autonomous decision makers. Section 4 considers self-coordination, by which the procedure for controlling fast timescale behavior with stabilizing feedback and risk-averse decision policies addresses multi-level performance robustness. Section 5 also presents a robust procedure for analyzing noncooperative modes of slow timescale interactions and for designing Nash equilibrium actions. Finally, a summary and remarks are given in Sect. 6.

2 Setting the Scene

Before going into a formal presentation, it is necessary to consider some conceptual notations. To be specific, for a given Hilbert space X with norm $||\cdot||_X$, $1 \leq p \leq \infty$, and $a, b \in \mathbb{R}$ such that $a \leq b$, a Banach space is defined as follows $L^p_{\mathcal{F}}(a, b; X) \triangleq \{\phi(\cdot) = \{\phi(t, \omega) : a \leq t \leq b\}$ such that $\phi(\cdot)$ is an X-valued $\mathcal{F}_t$-measurable process on $[a, b]$ with $E\{\int_a^b ||\phi(t, \omega)||^p_X \mathrm{d}t\} < \infty\}$ with the norm $||\phi(\cdot)||_{\mathcal{F},p} \triangleq (E\{\int_a^b ||\phi(t, \omega)||^p_X \mathrm{d}t\})^{1/p}$, where the elements ω of the filtered sigma field $\mathcal{F}_t$ of a sample description space Ω that is adapted for the time horizon $[a, b]$ are random outcomes or events. Also, the Banach space of X-valued continuous functionals on $[a, b]$ with the max-norm induced by $||\cdot||_X$ is denoted by $C(a, b; X)$. The deterministic version and its associated norm are written as $L^p(a, b; X)$ and $||\cdot||_p$.

To understand the evolutions of complex systems, system dynamic approaches and models are excellent tools for simulating and exploring evolving processes. Herein, a strongly coupled slow-core process with the initial-value state $x_0(t_0) = x_{00}$

$$\mathrm{d}x_0(t) = \left(A_0 x_0(t) + \sum_{j=1}^{N} A_{0j} x_j(t) + \sum_{j=1}^{N} B_{0j} u_j(t) \right) \mathrm{d}t + G_0 \mathrm{d}w(t)\,, \qquad (1)$$

where the constant coefficients $A_0 \in \mathbb{R}^{n_0 \times n_0}$, $A_{0j} \in \mathbb{R}^{n_0 \times n_j}$, $B_{0j} \in \mathbb{R}^{n_0 \times m_j}$, $G_0 \in \mathbb{R}^{n_0 \times p_0}$, while N weakly coupled fast-core processes with the constant coefficients $A_{i0} \in \mathbb{R}^{n_i \times n_0}$, $A_{ii} \in \mathbb{R}^{n_i \times n_i}$, $A_{ij} \in \mathbb{R}^{n_i \times n_j}$, $B_{ii} \in \mathbb{R}^{m_i \times m_i}$, and $G_i \in \mathbb{R}^{n_i \times p_0}$

$$\varepsilon_i \mathrm{d}x_i(t) = \left(A_{i0}x_0(t) + A_{ii}x_i(t) + \sum_{j=1, j\neq i}^{N} \varepsilon_{ij} A_{ij} x_j(t) + B_{ii}u_i(t) \right) \mathrm{d}t + \sqrt{\varepsilon_i} G_i \mathrm{d}w(t)$$

$$x_i(t_0) = x_{i0}, \quad i = 1, \ldots, N \tag{2}$$

are proposed to account for the pairing of mutual influence with temporal features, by which N decision makers are now capable of dynamically coordinating their activities and cooperating with others. Each decision maker i is assumed to be acting autonomously and so making decisions about what to do at engagement time through information sampling and exchanges available locally

$$\mathrm{d}y_{0i}(t) = (C_{0i}x_0(t) + C_i x_i(t))\,\mathrm{d}t + \mathrm{d}v_{0i}(t), \quad i = 1, \ldots, N \tag{3}$$

$$\mathrm{d}y_{ii}(t) = \left(\sqrt{\varepsilon_i} C_{i0}x_0(t) + C_{ii}x_i(t)\right)\mathrm{d}t + \sqrt{\varepsilon_i}\mathrm{d}v_{ii}(t)\,. \tag{4}$$

Furthermore, all decision makers operate within local environments modeled by the filtered probability spaces that are defined with p_0, q_{0i}, and q_{ii}-dimensional stationary Wiener processes adapted for $[t_0, t_f]$ together with the correlations of independent increments for all $\tau, \xi \in [t_0, t_f]$

$$E\left\{[w(\tau) - w(\xi)][w(\tau) - w(\xi)]^{\mathrm{T}}\right\} = W|\tau - \xi|, \quad W > 0$$

$$E\left\{[v_{0i}(\tau) - v_{0i}(\xi)][v_{0i}(\tau) - v_{0i}(\xi)]^{\mathrm{T}}\right\} = V_{0i}|\tau - \xi|, \quad V_{0i} > 0$$

$$E\left\{[v_{ii}(\tau) - v_{ii}(\xi)][v_{ii}(\tau) - v_{ii}(\xi)]^{\mathrm{T}}\right\} = V_{ii}|\tau - \xi|, \quad V_{ii} > 0.$$

The small singular perturbation parameters $\varepsilon_i > 0$ for $i = 1, \ldots, N$ represent different time constants, masses, etc., which help to account for decision makers' responsiveness to internal and external changes from their environments as well as their inertia and stability over time. Other small regular perturbation parameters ε_{ij} are weak coupling between the decision makers.

Note that each decision maker now has his/her own observations (3) and (4), makes his/her own admissible decisions $u_i \in U_i \subseteq L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{m_i})$, and has his/her own unique history of interactions (2) with fast states $x_i \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{n_i})$. The coefficient matrices A_{ii} are also assumed to be invertible. For a practical purpose, the $\sqrt{\varepsilon_i}$ factor is further inserted to both process and observation noise terms to ensure the fast variables x_i physically meaningful for control and estimation purposes. A more complete discussion about the use and justification of this practice can be found in [2,3].

3 Multi-Model Generation

So far, there has been nothing of how decision makers can work together. In this section, decision makers in the complex system are active in interpreting events and subsequently motivating appropriate responses to these events. In this sense, decision maker i decides to continue or discontinue relations with others. For example, decision maker i may choose to neglect the fast dynamics of decision makers j and the weak interconnections between the fast timescale processes; for example, by setting $\varepsilon_j = 0$ on the left hand side of (2) and $\varepsilon_{ij} = 0$ in (2). For any $j = 1, \ldots, N$ and $j \neq i$, the long-term behavior or steady-state dynamics of neighboring decision makers j is then given by

$$\overline{x}_j(t)\mathrm{d}t = -A_{jj}^{-1}\left(\left(A_{j0}x_0(t) + B_{jj}u_j(t)\right)\mathrm{d}t + \sqrt{\varepsilon_j}G_j\mathrm{d}w(t)\right). \tag{5}$$

As mentioned in [3], the result (5) turns out to be as valid inputs to the slow timescale process (1). Viewed from the mutual influence of one decision maker to those of others, self-coordination preferred by decision maker i is hence described by a simplified model whose dynamical states $x_0^i \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{n_0})$ and $x_i^i \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{n_i})$, resulted from the substitution of the stochastic process (5) into those of (1) and (2)

$$\mathrm{d}x_0^i(t) = \left(A_0^i x_0^i(t) + A_{0i}x_i^i(t) + \sum_{j=1, j\neq i}^{N} B_{0j}^i u_j(t) + B_{0i}u_i(t)\right)\mathrm{d}t + G_0^i\mathrm{d}w(t) \tag{6}$$

$$\varepsilon_i\mathrm{d}x_i^i(t) = \left(A_{i0}x_0^i(t) + A_{ii}x_i^i(t) + B_{ii}u_i(t)\right)\mathrm{d}t + \sqrt{\varepsilon_i}G_i\mathrm{d}w(t), \tag{7}$$

where the initial-value states $x_0^i(t_0) \equiv x_{00}$ and $x_i^i(t_0) \equiv x_{i0}$, while the coefficients are given by $A_0^i \triangleq A_0 - \sum_{j=1, j\neq i}^{N} A_{0j}A_{jj}^{-1}A_{j0}$, $B_{0j}^i \triangleq B_{0j} - A_{0j}A_{jj}^{-1}B_{jj}$, and $G_0^i \triangleq G_0 - \sum_{j=1, j\neq i}^{N}\sqrt{\varepsilon_j}A_{0j}A_{jj}^{-1}G_j$. With the simplified model (6) and (7) in mind, decision maker i can bring his/her activities into coordination with the activities of others via his/her aggregate observations $y_i^i \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{q_{i0}+q_{ii}})$

$$\begin{aligned}\mathrm{d}y_i^i(t) &= \begin{bmatrix} C_{0i} & C_i \\ C_{i0} & \frac{1}{\sqrt{\varepsilon_i}}C_{ii}\end{bmatrix}\begin{bmatrix} x_0^i(t) \\ x_i^i(t)\end{bmatrix}\mathrm{d}t + \mathrm{d}v_i(t) \\ &= \left(C_{is}x_0^i(t) + D_{is}u_i(t)\right)\mathrm{d}t + \mathrm{d}v_{is}(t), \quad i = 1, \ldots, N\end{aligned} \tag{8}$$

provided that $\mathrm{d}y_i^i \triangleq \begin{bmatrix} \mathrm{d}y_{0i} \\ \frac{1}{\sqrt{\varepsilon_i}}\mathrm{d}y_{ii}\end{bmatrix}$, $\mathrm{d}v_i \triangleq \begin{bmatrix}\mathrm{d}v_{0i} \\ \mathrm{d}v_{ii}\end{bmatrix}$, $\mathrm{d}v_{is} \triangleq \begin{bmatrix}\mathrm{d}v_{0i} - \sqrt{\varepsilon_i}C_iA_{ii}^{-1}G_i\mathrm{d}w \\ \mathrm{d}v_{ii} - C_{ii}A_{ii}^{-1}G_i\mathrm{d}w\end{bmatrix}$, $C_{is} \triangleq \begin{bmatrix} C_{0i} - C_iA_{ii}^{-1}A_{i0} \\ C_{i0} - \frac{1}{\sqrt{\varepsilon_i}}C_{ii}A_{ii}^{-1}A_{i0}\end{bmatrix}$, and $D_{is} \triangleq \begin{bmatrix} -C_iA_{ii}^{-1}B_{ii} \\ -\frac{1}{\sqrt{\varepsilon_i}}C_{ii}A_{ii}^{-1}B_{ii}\end{bmatrix}$.

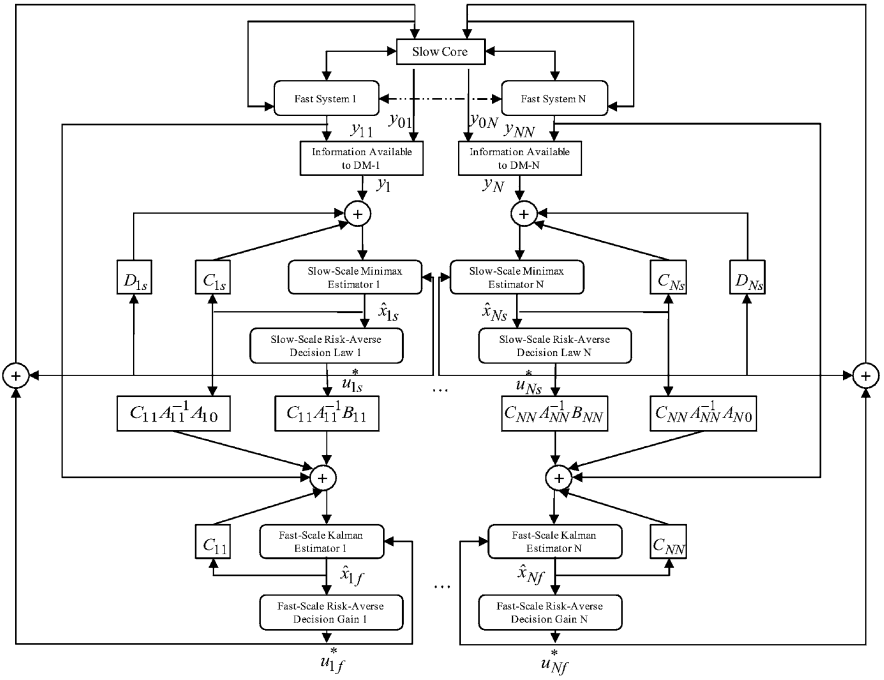

Fig. 1 A two-level structure for online dynamic coordination

Under such the simplified model (6) and (7) being used by those decision makers, the entire group of N decision makers may work as a team with each one fitting in, where he/she thinks his/her effort will be most effective and will interfere least with the others. Occasionally decision makers interact at the slow timescale level; but most of the adjustments take place silently and without deliberation at the fast timescale levels. All these situations, where self-coordination is possible, require that each decision maker be able to possess the knowledge of the parameters associated with the simplified model (6) and (7). With references to the work [1], a two-level structure, as shown in Fig. 1 for online dynamic coordination, is adapted with appropriate paradigms for estimate observations and decision making with risk aversion. In particular, the individual assessment of the alternatives available is obtained by solving $2N$ low-order problems: N independent optimal statistical control problems for each decision maker at the fast timescale level; and N constrained stochastic Nash games at the slow timescale level.

4 Fast Interactions

Short horizon interactions are now concerned with establishing a framework for information analysis and performance-risk bearing decisions that permit consequences anticipated on performance reliability for the decision makers in charge of local fast operations within stochastic environments

$$\varepsilon_i \mathrm{d}x_{if}(t) = \big(A_{ii}x_{if}(t) + B_{ii}u_{if}\big)\,\mathrm{d}t + \sqrt{\varepsilon_i}G_i\mathrm{d}w(t), \quad x_{if}(t_0) = x_{i0} \tag{9}$$

$$\mathrm{d}y_{iif} = C_{ii}x_{if}(t)\mathrm{d}t + \mathrm{d}v_{ii}(t), \quad i = 1,\ldots,N \quad f \sim \text{fast}. \tag{10}$$

Yet, the decision makers attempt to make risk-bearing decisions u_{if} from the admissible sets $U_{if} \subset L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{m_i})$ for reliable attainments of integral-quadratic utilities; for instance, $J_{if} : \mathbb{R}^{n_i} \times U_{if} \mapsto \mathbb{R}^+$ with the rules of action

$$J_{if}(x_{i0}, u_{if}) = \varepsilon_i x^T_{if}(t_f)Q^f_{if}x_{if}(t_f) + \int_{t_0}^{t_f}\Big[x^T_{if}(\tau)Q_{if}x_{if}(\tau) + u^T_{if}(\tau)R_{if}u_{if}(\tau)\Big]\mathrm{d}\tau\,. \tag{11}$$

The design-weighting matrices $Q^f_{if} \in \mathbb{R}^{n_i\times n_i}$, $Q_{if} \in \mathbb{R}^{n_i\times n_i}$, and $R_{if} \in \mathbb{R}^{m_i\times m_i}$ are real, symmetric, and positive semidefinite with R_{if} invertible. The relative "size" of Q_{if} and R_{if} enforces trade-offs between the speed of response and the size of the control decision. At this point, it is convenient to use the Kalman-like estimates $\hat{x}_{if} \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{n_i})$ with the initial state estimates $\hat{x}_{if}(t_0) = x_{i0}$, Kalman gain $L_{if}(t) \triangleq P_{if}(t)C^T_{ii}V^{-1}_{ii}$, and the estimate-error covariances $P_{if} \in C^1(t_0, t_f; \mathbb{R}^{n_i\times n_i})$ with the initial-value conditions $P_{if}(t_0) = 0$ for $i = 1,\ldots,N$

$$\varepsilon_i \mathrm{d}\hat{x}_{if}(t) = \big(A_{ii}\hat{x}_{if}(t) + B_{ii}u_{if}(t)\big)\,\mathrm{d}t + P_{if}(t)C^T_{ii}V^{-1}_{ii}(\mathrm{d}y_{iif}(t) - C_{ii}\hat{x}_{if}(t)\mathrm{d}t) \tag{12}$$

$$\varepsilon_i \frac{\mathrm{d}}{\mathrm{d}t}P_{if}(t) = P_{if}(t)A^T_{ii} + A_{ii}P_{if}(t) - P_{if}(t)C^T_{ii}V^{-1}_{ii}C_{ii}P_{if}(t) + G_iWG^T_i \tag{13}$$

to approximately describe the future evolution of the fast timescale process (9) when different control decision processes applied.

From (13), the covariance of error estimates is independent of decision action and observations. Therefore, to parameterize the conditional densities $p(x_{if}(t)|\mathcal{F}_t)$ and $i = 1,\ldots,N$, the conditional mean $\hat{x}_{if}(t)$ minimizing error-estimate covariance of $x_{if}(t)$ is only needed. Thus, a family of decision policies is chosen of the form: $\gamma_{if} : \Gamma_{if} \mapsto U_{if}$, $u_{if} = \gamma_{if}(\eta_{if})$, and $\eta_{if} \triangleq \big(t, \hat{x}_{if}(t)\big)$. Since the quadratic decision problem (9) and (11) is of interest, the search for closed-loop feedback decision laws is then restricted within the strategy space, which permits a linear feedback synthesis

$$u_{if}(t) \triangleq K_{if}(t)\hat{x}_{if}(t)\,, \quad \text{for } i = 1,\ldots,N \tag{14}$$

wherein the elements of $K_{if}(t) \in C(t_0, t_f; \mathbb{R}^{m_i \times n_i})$ represent admissible fast timescale decision gains defined in some appropriate sense.

Moreover, the pairs (A_{ii}, B_{ii}) and (A_{ii}, C_{ii}) for $i = 1, \ldots, N$ are assumed to be stabilizable and detectable, respectively. Under this assumption, such feedback and filter gains $K_{if}(\cdot)$ and $L_{if}(\cdot)$ exist so that the aggregate system dynamics is exponentially stable.

The following result provides a representation of riskiness from the standpoint of higher-order characteristics pertaining to probabilistic performance distributions with respect to the underlying stochastic environment. This representation also has significance at the level of decision making, where risk-averse courses of action originate.

Theorem 1 (Fast Interactions—Performance-measure Statistics). *For fast interactions governed by (9) and (11), the pairs (A_{ii}, B_{ii}) and (A_{ii}, C_{ii}) are stabilizable and detectable. Then, for any given $k_{if} \in \mathbb{N}$, the k_{if}th cumulant associated with the performance-measure (11) for decision maker i is given as follows:*

$$\kappa_{if}^{k_{if}} = x_{i0}^T H_{if}^{11}(t_0, k_{if}) x_{i0} + D_{if}(t_0, k_{if}), \quad i = 1, \ldots, N, \tag{15}$$

where all the cumulant variables $\{H_{if}^{11}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{12}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{21}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{22}(\alpha, r)\}_{r=1}^{k_{if}}$ and $\{D_{if}(\alpha, r)\}_{r=1}^{k_{if}}$ evaluated at $\alpha = t_0$ satisfy the matrix and scalar-valued differential equations (with the dependence of $H_{if}^{11}(\alpha, r)$, $H_{if}^{12}(\alpha, r)$, $H_{if}^{21}(\alpha, r)$, $H_{if}^{22}(\alpha, r)$, and $D_{if}(\alpha, r)$ upon the admissible K_{if} suppressed)

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} H_{if}^{11}(\alpha, r) = F_{if,r}^{11}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{21}(\alpha), K_{if}(\alpha)) \tag{16}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} H_{if}^{12}(\alpha, r) = F_{if,r}^{12}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{22}(\alpha), K_{if}(\alpha)) \tag{17}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} H_{if}^{21}(\alpha, r) = F_{if,r}^{21}(\alpha, H_{if}^{11}(\alpha), H_{if}^{21}(\alpha), H_{if}^{22}(\alpha), K_{if}(\alpha)) \tag{18}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} H_{if}^{22}(\alpha, r) = F_{if,r}^{22}(\alpha, H_{if}^{12}(\alpha), H_{if}^{21}(\alpha), H_{if}^{22}(\alpha)) \tag{19}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} D_{if}(\alpha, r) = G_{if,r}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{21}(\alpha), H_{if}^{22}(\alpha)), \tag{20}$$

where the terminal-value conditions $H_{if}^{11}(t_f, 1) = \varepsilon_i Q_{if}^f$, $H_{if}^{11}(t_f, r) = 0$ for $2 \leq r \leq k_{if}$; $H_{if}^{12}(t_f, 1) = \varepsilon_i Q_{if}^f$, $H_{if}^{12}(t_f, r) = 0$ for $2 \leq r \leq k_{if}$; $H_{if}^{21}(t_f, 1) = \varepsilon_i Q_{if}^f$, $H_{if}^{21}(t_f, r) = 0$ for $2 \leq r \leq k_{if}$; $H_{if}^{22}(t_f, 1) = \varepsilon_i Q_{if}^f$, $H_{if}^{22}(t_f, r) = 0$ for $2 \leq r \leq k_{if}$; and $D_{if}(t_f, r) = 0$ for $1 \leq r \leq k_{if}$. Furthermore, all the k_{if}-tuple variables $H_{if}^{11}(\alpha) \triangleq (H_{if}^{11}(\alpha, 1), \ldots, H_{if}^{11}(\alpha, k_{if}))$; $H_{if}^{12}(\alpha) \triangleq (H_{if}^{12}(\alpha, 1), \ldots, H_{if}^{12}(\alpha, k_{if}))$; $H_{if}^{21}(\alpha) \triangleq (H_{if}^{21}(\alpha, 1), \ldots, H_{if}^{21}(\alpha, k_{if}))$; and $H_{if}^{22}(\alpha) \triangleq (H_{if}^{22}(\alpha, 1), \ldots, H_{if}^{22}(\alpha, k_{if}))$.

Proof. With the interest of space limitation, the proof is omitted. Interested readers are referred to the Appendix and [4] for the mathematical definitions of the mappings governing the right members of (16)–(20) and in-depth development, respectively.

To anticipate for a well-posed optimization problem that follows, some sufficient conditions for the existence of solutions to the cumulant-generating equations (16)–(20) in the calculation of performance-measure statistics are now presented in the sequel.

Theorem 2 (Fast Interactions—Existence of Performance-Measure Statistics). *Let* (A_{ii}, B_{ii}) *and* (A_{ii}, C_{ii}) *be stabilizable and detectable. Then, any given any given* $k_{if} \in \mathbb{N}$*, the time-backward matrix and scalar-valued differential equations (16)–(20) admit unique and bounded solutions* $\{H_{if}^{11}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{12}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{21}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{22}(\alpha, r)\}_{r=1}^{k_{if}}$, *and* $\{D_{if}(\alpha, r)\}_{r=1}^{k_{if}}$ *on* $[t_0, t_f]$.

Proof. With references to stabilizable and detectable assumptions, there always exist some feedback decision gains $K_{if} \in C(t_0, t_f; \mathbb{R}^{m_i \times n_i})$ and filter gains $L_{if} \in C(t_0, t_f; \mathbb{R}^{n_i \times q_{ii}})$ so that composite state matrices $F_{A_{ii}+B_{ii}K_{if}} \in C(t_0, t_f; \mathbb{R}^{n_i \times n_i})$ and $F_{A_{ii}-L_{if}C_{ii}} \in C(t_0, t_f; \mathbb{R}^{n_i \times n_i})$ are exponentially stable on $[t_0, t_f]$. Therefore, the state transition matrices $\Phi^{if}_{A_{ii}+B_{ii}K_{if}}(t, t_0)$ and $\Phi^{if}_{A_{ii}-L_{if}C_{ii}}(t, t_0)$ associated with $F_{A_{ii}+B_{ii}K_{if}}(t)$ and $F_{A_{ii}-L_{if}C_{ii}}(t)$ have the properties: $\lim_{t_f \to \infty} ||\Phi^{if}_{A_{ii}+B_{ii}K_{if}}(t_f, \sigma)|| = 0$ and $\lim_{t_f \to \infty} \int_{t_0}^{t_f} ||\Phi^{if}_{A_{ii}+B_{ii}K_{if}}(t_f, \sigma)||^2 d\sigma < \infty$. By the matrix variation of constant formula, the unique and time-continuous solutions to the (16)–(20) can be expressed in terms of $\Phi^{if}_{A_{ii}+B_{ii}K_{if}}(t, t_0)$ and $\Phi^{if}_{A_{ii}-L_{if}C_{ii}}(t, t_0)$. As long as the growth rate of the integrals is not faster than the exponentially decreasing rate of two factors of $\Phi^{if}_{A_{ii}+B_{ii}K_{if}}(t, t_0)$ and $\Phi^{if}_{A_{ii}-L_{if}C_{ii}}(t, t_0)$, it is then concluded that there exist upper bounds on the unique and time-continuous solutions $\{H_{if}^{11}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{12}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{21}(\alpha, r)\}_{r=1}^{k_{if}}$, $\{H_{if}^{22}(\alpha, r)\}_{r=1}^{k_{if}}$, and $\{D_{if}(\alpha, r)\}_{r=1}^{k_{if}}$ for any time interval $[t_0, t_f]$.

Remark 1. Notice that the solutions $H_{if}^{11}(\alpha)$, $H_{if}^{12}(\alpha)$, $H_{if}^{21}(\alpha)$, $H_{if}^{22}(\alpha)$, and $D_{if}(\alpha)$ of the (16)–(20) depend on the admissible decision gain K_{if} of the feedback decision law (14) by decision makers i for $i = 1, \ldots, N$. In the sequel and elsewhere, when this dependence is needed to be clear, then the notations $H_{if}^{11}(\alpha, K_{if})$, $H_{if}^{12}(\alpha, K_{if})$, $H_{if}^{21}(\alpha, K_{if})$, $H_{if}^{22}(\alpha, K_{if})$, and $D_{if}(\alpha, K_{if})$ should be used to denote the solution trajectories of the dynamics (16)–(20) with the given feedback decision gain K_{if}.

Next, the components of $4k_{if}$-tuple $\mathcal{H}_{if}$ and k_{if}-tuple $\mathcal{D}_{if}$ variables are defined by

$$\begin{aligned}\mathcal{H}_{if} &\triangleq \left(\mathcal{H}_{if}^{1}, \ldots, \mathcal{H}_{if}^{k_{if}}, \mathcal{H}_{if}^{k_{if}+1} \ldots, \mathcal{H}_{if}^{2k_{if}}, \mathcal{H}_{if}^{2k_{if}+1}, \ldots, \mathcal{H}_{if}^{3k_{if}}, \mathcal{H}_{if}^{3k_{if}+1}, \ldots, \mathcal{H}_{if}^{4k_{if}}\right) \\ &= (H_{if}^{11}(\cdot, 1), \ldots, H_{if}^{11}(\cdot, k_{if}), H_{if}^{12}(\cdot, 1), \ldots, H_{if}^{12}(\cdot, k_{if}), H_{if}^{21}(\cdot, 1), \ldots, H_{if}^{22}(\cdot, k_{if}))\end{aligned}$$

and

$$\begin{aligned}\mathcal{D}_{if} &\triangleq (\mathcal{D}_{if}^{1}, \ldots, \mathcal{D}_{if}^{k_{if}}) \\ &= (D_{if}(\cdot, 1), \ldots, D_{if}(\cdot, k_{if})).\end{aligned}$$

Henceforth, the product systems of dynamical equations (16)–(20), whose respective mappings $F_{if}^{11} \triangleq F_{if,1}^{11} \times \cdots \times F_{if,k_{if}}^{11}$, $F_{if}^{12} \triangleq F_{if,1}^{12} \times \cdots \times F_{if,k_{if}}^{12}$, $F_{if}^{21} \triangleq F_{if,1}^{21} \times \cdots \times F_{if,k_{if}}^{21}$, $F_{if}^{22} \triangleq F_{if,1}^{22} \times \cdots \times F_{if,k_{if}}^{22}$, and $G_{if} \triangleq G_{if,1} \times \cdots \times G_{if,k_{if}}$ are defined on $[t_0, t_f] \times (\mathbb{R}^{n_i \times n_i})^{4k_{if}} \times \mathbb{R}^{m_i \times n_i}$ and $[t_0, t_f] \times (\mathbb{R}^{n_i \times n_i})^{4k_{if}}$ in the optimal statistical control with dynamical output feedback, become

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}(\alpha) = \mathcal{F}_{if}(\alpha, \mathcal{H}_{if}(\alpha), K_{if}(\alpha)), \qquad \mathcal{H}_{if}(t_f) = \mathcal{H}_{if}^{f}, \tag{21}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{D}_{if}(\alpha) = \mathcal{G}_{if}(\alpha, \mathcal{H}_{if}(\alpha)), \qquad \mathcal{D}_{if}(t_f) = \mathcal{D}_{if}^{f}, \tag{22}$$

under the definition $\mathcal{F}_{if} \triangleq F_{if}^{11} \times F_{if}^{12} \times F_{if}^{21} \times F_{if}^{22}$ together with the aggregate terminal-value conditions

$$\mathcal{H}_{if}^{f} \triangleq \varepsilon_i Q_{if}^{f} \times \underbrace{0 \times \cdots \times 0}_{(k_{if}-1)\text{-times}} \times \varepsilon_i Q_{if}^{f} \times \underbrace{0 \times \cdots \times 0}_{(k_{if}-1)\text{-times}} \times \cdots \times \varepsilon_i Q_{if}^{f} \times \underbrace{0 \times \cdots \times 0}_{(k_{if}-1)\text{-times}}$$

$$\mathcal{D}_{if}^{f} \triangleq \underbrace{0 \times \cdots \times 0}_{k_{if}\text{-times}}.$$

Given the evidences on surprises of utilities and preferences that now support the knowledge and beliefs of performance riskiness, all decision makers hence form rational expectations about the future and make decisions on the basis of this knowledge and these beliefs.

Definition 1 (Fast Interactions—Risk-Value Aware Performance Index). As defined herein, the optimal statistical control consists in determining risk-averse decision u_{if} to minimize the new performance index ϕ_{if}^{0}, which is defined on a subset of $\{t_0\} \times (\mathbb{R}^{n_i \times n_i})^{k_{if}} \times (\mathbb{R}^{k_{if}})$ such that

$$\phi_{if}^{0} \triangleq \underbrace{\mu_{if}^{1}\kappa_{if}^{1}}_{\text{Standard Measure}} + \underbrace{\mu_{if}^{2}\kappa_{if}^{2} + \cdots + \mu_{if}^{k_{if}}\kappa_{if}^{k_{if}}}_{\text{Risk Measures}}, \tag{23}$$

where the rth order performance-measure statistics $\kappa_{if}^{r} \equiv \kappa_{if}^{r}(t_0, \mathcal{H}_{if}^{r}(t_0), \mathcal{D}_{if}^{r}(t_0)) = x_{i0}^{T}\mathcal{H}_{if}^{r}(t_0)x_{i0} + \mathcal{D}_{if}^{r}(t_0)$ for $1 \leq r \leq k_{if}$ and the sequence $\mu^{if} = \{\mu_{if}^{r} \geq 0\}_{r=1}^{k_{if}}$ with $\mu_{if}^{1} > 0$. Parametric design measures μ_{if}^{r} considered here represent different emphases on higher-order statistics and prioritizations by decision maker i toward robust performance and risk sensitivity.

Remark 2. This multi-objective performance index is interpreted as a linear combination of the first k_{if} performance-measure statistics of the integral-quadratic utility (11), on the one hand, and a value and risk model, on the other, to reflect the trade-offs between performance benefits and risks.

From the above definition, it is clear that the statistical problem is an initial-cost problem, in contrast with the more traditional terminal-cost class of investigations. One may address an initial cost problem by introducing changes of variables, which convert it to a terminal-cost problem. This modifies the natural context of the optimal statistical control, however, which it is preferable to retain. Instead, one may take a more direct dynamic programming approach to the initial-cost problem. Such an approach is illustrative of the more general concept of the principle of optimality, an idea tracing its roots back to the seventeenth century. The development in the sequel is motivated by the excellent treatment in [5] and is intended to follow it closely. Because [5] embodies the traditional endpoint problem and corresponding use of dynamic programming, it is necessary to make appropriate modifications in the sequence of results, as well as to introduce the terminology of the optimal statistical control.

Let the terminal time t_f and states $(\mathcal{H}_{if}^f, \mathcal{D}_{if}^f)$ be given. Then, the other end condition involved the initial time t_0 and state pair $(\mathcal{H}_{if}^0, \mathcal{D}_{if}^0)$ are specified by a target set requirement.

Definition 2 (Fast Interactions—Target Sets). $(t_0, \mathcal{H}_{if}^0, \mathcal{D}_{if}^0) \in \hat{\mathcal{M}}_{if}$, where the target set $\hat{\mathcal{M}}_{if}$ and $i = 1, \ldots, N$, is a closed subset defined by $[t_0, t_f] \times (\mathbb{R}^{n_i \times n_i})^{4k_{if}} \times \mathbb{R}^{k_{if}}$.

For the given terminal data $(t_f, \mathcal{H}_{if}^f, \mathcal{D}_{if}^f)$, the class $\hat{\mathcal{K}}^{if}_{t_f, \mathcal{H}_{if}^f, \mathcal{D}_{if}^f; \mu^{if}}$ of admissible feedback gain is defined as follows.

Definition 3 (Fast Interactions—Admissible Fedback Gains). Let the compact subset $\overline{\mathcal{K}}_{if} \subset \mathbb{R}^{m_i \times n_i}$ be the set of allowable gain values. For the given $k_{if} \in \mathbb{N}$ and the sequence $\mu^{if} = \{\mu_{if}^r \geq 0\}_{r=1}^{k_{if}}$ with $\mu_{if}^1 > 0$, let $\hat{\mathcal{K}}^{if}_{t_f, \mathcal{H}_{if}^f, \mathcal{D}_{if}^f; \mu^{if}}$ be the class of $\mathcal{C}([t_0, t_f]; \mathbb{R}^{m_i \times n_i})$ with values $K_{if}(\cdot) \in \overline{\mathcal{K}}_{if}$, for which the performance index (23) is finite and for which the trajectory solutions to the dynamic equations (21) and (22) reach $(t_0, \mathcal{H}_{if}^0, \mathcal{D}_{if}^0) \in \hat{\mathcal{M}}_{if}$.

Now, the optimization problem is to minimize the risk-value aware performance index (23) over all admissible feedback gains $K_{if} = K_{if}(\cdot)$ in $\hat{\mathcal{K}}^{if}_{t_f, \mathcal{H}_{if}^f, \mathcal{D}_{if}^f; \mu^{if}}$.

Definition 4 (Fast Interactions—Optimization of Mayer Problem). Suppose that $k_{if} \in \mathbb{N}$ and the sequence $\mu^{if} = \{\mu_{if}^r \geq 0\}_{r=1}^{k_{if}}$ with $\mu_{if}^1 > 0$ are fixed. Then, the control optimization with output-feedback information pattern is given by

$$\min_{K_{if}(\cdot) \in \hat{\mathcal{K}}^{if}_{t_f, \mathcal{H}_{if}^f, \mathcal{D}_{if}^f; \mu^{if}}} \phi_{if}^0 \left(t_0, \mathcal{H}_{if}(t_0, K_{if}), \mathcal{D}_{if}(t_0, K_{if}) \right),$$

subject to the dynamical equations (21) and (22) on $[t_0, t_f]$.

It is important to recognize that the optimization considered here is in Mayer form and can be solved by applying an adaptation of the Mayer form verification theorem of dynamic programming as given in [5]. To embed the aforementioned optimization into a larger optimal control problem, the terminal time and states $(t_f, \mathcal{H}_{if}^f, \mathcal{D}_{if}^f)$ are parameterized as $(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$. Thus, the value function for this optimization problem is now depending on the terminal condition parameterizations.

Definition 5 (Fast Interactions—Value Function). Suppose that $(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) \in [t_0, t_f] \times (\mathbb{R}^{n_i \times n_i})^{4k_{if}} \times \mathbb{R}^{k_{if}}$ is given and fixed. Then, the value function $\mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$ is defined by

$$\mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) \triangleq \inf_{K_{if}(\cdot) \in \hat{\mathcal{K}}^{if}_{\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}; \mu^{if}}} \phi_{if}^0 \left(t_0, \mathcal{H}_{if}(t_0, K_{if}), \mathcal{D}_{if}(t_0, K_{if})\right).$$

For convention, $\mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) \triangleq \infty$ when $\hat{\mathcal{K}}^{if}_{\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}; \mu^{if}}$ is empty. To avoid cumbersome notation, the dependence of trajectory solutions on $K_{if}(\cdot)$ is suppressed. Next, some candidates for the value function are constructed with the help of the concept of reachable set.

Definition 6 (Fast Interactions—Reachable Set). Let the reachable set $\hat{\mathcal{Q}}_{if}$ and $i = 1, \ldots, N$ be

$$\hat{\mathcal{Q}}_{if} \triangleq \left\{(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) \in [t_0, t_f] \times (\mathbb{R}^{n_i \times n_i})^{4k_{if}} \times \mathbb{R}^{k_{if}} \,:\, \hat{K}^{if}_{\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}; \mu^{if}} \neq \emptyset\right\}.$$

Notice that $\hat{\mathcal{Q}}_{if}$ contains a set of points $(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$, from which it is possible to reach the target set $\hat{\mathcal{M}}_{if}$ with some trajectory pairs corresponding to a continuous decision gain. Furthermore, the value function must satisfy both a partial differential inequality and an equation at each interior point of the reachable set, at which it is differentiable.

Theorem 3 (Fast Interactions—Hamilton–Jacobi–Bellman (HJB) Equation). *Let $(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$ be any interior point of the reachable set $\hat{\mathcal{Q}}_{if}$, at which the scalar-valued function $\mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$ is differentiable. Then $\mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$ satisfies the partial differential inequality*

$$\begin{aligned} 0 \geq\ & \frac{\partial}{\partial \varepsilon} \mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) + \frac{\partial}{\partial \mathrm{vec}(\mathcal{Y}_{if})} \mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) \mathrm{vec}(\mathcal{F}_{if}(\varepsilon, \mathcal{Y}_{if}, K_{if})) \\ & + \frac{\partial}{\partial \mathrm{vec}(\mathcal{Z}_{if})} \mathcal{V}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) \mathrm{vec}(\mathcal{G}_{if}(\varepsilon, \mathcal{Y}_{if})) \end{aligned} \tag{24}$$

for all $K_{if} \in \overline{\mathcal{K}}_{if}$ and $\mathrm{vec}(\cdot)$ *the vectorizing operator of enclosed entities.*

If there is an optimal feedback decision gain K_{if}^* *in* $\hat{\mathcal{K}}^{if}_{\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if};\mu^{if}}$, *then the partial differential equation of dynamic programming*

$$0 = \min_{K_{if}\in\overline{\mathcal{K}}_{if}} \left\{ \frac{\partial}{\partial \text{vec}(\mathcal{Y}_{if})}\mathcal{V}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if})\text{vec}(\mathcal{F}_{if}(\varepsilon,\mathcal{Y}_{if},K_{if})) \right.$$
$$\left. + \frac{\partial}{\partial \text{vec}(\mathcal{Z}_{if})}\mathcal{V}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if})\text{vec}(\mathcal{G}_{if}(\varepsilon,\mathcal{Y}_{if})) + \frac{\partial}{\partial\varepsilon}\mathcal{V}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if}) \right\} \tag{25}$$

is satisfied. The minimum in (25) is achieved by the optimal feedback decision gain $K_{if}^*(\varepsilon)$ *at* ε.

Proof. Interested readers are referred to the mathematical details in [6].

The verification theorem in the optimal statistical control notation is stated as follows.

Theorem 4 (Fast Interactions—Verification Theorem). *Fix* $k_{if} \in \mathbb{N}$ *and let* $\mathcal{W}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if})$ *be a continuously differentiable solution of the HJB equation (25), which satisfies the boundary* $\mathcal{W}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if}) = \phi_{if}^0(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if})$ *for some* $(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if}) \in \hat{\mathcal{M}}_{if}$. *Let* $(t_f,\mathcal{H}_{if}^f,\mathcal{D}_{if}^f)$ *be a point of* $\hat{\mathcal{Q}}_{if}$, *let* K_{if} *be a feedback decision gain in* $\hat{\mathcal{K}}^{if}_{t_f,\mathcal{H}_{if}^f,\mathcal{D}_{if}^f;\mu^{if}}$ *and let* $\mathcal{H}_{if}$, $\mathcal{D}_{if}$ *be the corresponding solutions of the (21) and (22). Then,* $\mathcal{W}_{if}(\alpha,\mathcal{H}_{if}(\alpha),\mathcal{D}_{if}(\alpha))$ *is a non-increasing function of* α. *If* K_{if}^* *is a feedback decision gain in* $\hat{\mathcal{K}}^{if}_{t_f,\mathcal{H}_{if}^f,\mathcal{D}_{if}^f;\mu^{if}}$ *defined on* $[t_0,t_f]$ *with the corresponding solutions* $\mathcal{H}_{if}^*$ *and* $\mathcal{D}_{if}^*$ *of the preceding equations such that, for* $\alpha\in[t_0,t_f]$,

$$0 = \frac{\partial}{\partial\varepsilon}\mathcal{W}_{if}(\alpha,\mathcal{H}_{if}^*(\alpha),\mathcal{D}_{if}^*(\alpha))$$
$$+\frac{\partial}{\partial \text{vec}(\mathcal{Y}_{if})}\mathcal{W}_{if}(\alpha,\mathcal{H}_{if}^*(\alpha),\mathcal{D}_{if}^*(\alpha))\text{vec}(\mathcal{F}_{if}(\alpha,\mathcal{H}_{if}^*(\alpha),K_{if}^*(\alpha)))$$
$$+\frac{\partial}{\partial \text{vec}(\mathcal{Z}_{if})}\mathcal{W}_{if}(\alpha,\mathcal{H}_{if}^*(\alpha),\mathcal{D}_{if}^*(\alpha))\text{vec}(\mathcal{G}_{if}(\alpha,\mathcal{H}_{if}^*(\alpha)))\,, \tag{26}$$

then K_{if}^* *is an optimal feedback decision gain in* $\hat{\mathcal{K}}^{if}_{t_f,\mathcal{H}_{if}^f,\mathcal{D}_{if}^f;\mu^{if}}$ *and* $\mathcal{W}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if}) = \mathcal{V}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if})$, *where* $\mathcal{V}_{if}(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if})$ *is the value function.*

Proof. The detailed analysis can be found in the work by the first author [6].

Recall that the optimization problem being considered herein is in Mayer form, which can be solved by an adaptation of the Mayer form verification theorem. Thus, the terminal time and states $(t_f,\mathcal{H}_{if}^f,\mathcal{D}_{if}^f)$ are parameterized as $(\varepsilon,\mathcal{Y}_{if},\mathcal{Z}_{if})$ for a

family of optimization problems. For instance, the states (21) and (22) defined on the interval $[t_0, \varepsilon]$ now have terminal values denoted by $\mathcal{H}_{if}(\varepsilon) \equiv \mathcal{Y}_{if}$ and $\mathcal{D}_{if}(\varepsilon) \equiv \mathcal{Z}_{if}$, where $\varepsilon \in [t_0, t_f]$. Furthermore, with $k_{if} \in \mathbb{N}$ and $(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$ in $\hat{\mathcal{Q}}_{if}$, the following real-value candidate:

$$\mathcal{W}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) = x_{i0}^T \sum_{r=1}^{k_{if}} \mu_{if}^r (\mathcal{Y}_{if}^r + \mathcal{E}_{if}^r(\varepsilon)) x_{i0} + \sum_{r=1}^{k_{if}} \mu_{if}^r (\mathcal{Z}_{if}^r + \mathcal{T}_{if}^r(\varepsilon)) \quad (27)$$

for the value function is therefore differentiable. The time derivative of $\mathcal{W}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if})$ can also be shown of the form

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{W}_{if}(\varepsilon, \mathcal{Y}_{if}, \mathcal{Z}_{if}) = x_{i0}^T \sum_{r=1}^{k_{if}} \mu_{if}^r \left(\mathcal{F}_{if}^r(\varepsilon, \mathcal{Y}_{if}, K_{if}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{E}_{if}^r(\varepsilon) \right) x_{i0}$$
$$+ \sum_{r=1}^{k_{if}} \mu_{if}^r \left(\mathcal{G}_{if}^r(\varepsilon, \mathcal{Y}_{if}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{T}_{if}^r(\varepsilon) \right)$$

where the time parameter functions $\mathcal{E}_{if}^r \in \mathcal{C}^1([t_0, t_f]; \mathbb{R}^{n_i \times n_i})$ and $\mathcal{T}_{if}^r \in \mathcal{C}^1([t_0, t_r]; \mathbb{R})$ are to be determined.

At the boundary condition, it requires that

$$\mathcal{W}(t_0, \mathcal{Y}_{if}(t_0), \mathcal{Z}_{if}(t_0)) = \phi_{if}^0(t_0, \mathcal{Y}_{if}(t_0), \mathcal{Z}_{if}(t_0)),$$

which leads to

$$x_{i0}^T \sum_{r=1}^{k_{if}} \mu_{if}^r (\mathcal{Y}_{if}^r(t_0) + \mathcal{E}_{if}^r(t_0)) x_{i0} + \sum_{r=1}^{k_{if}} \mu_{if}^r (\mathcal{Z}_{if}^r(t_0) + \mathcal{T}_{if}^r(t_0))$$
$$= x_{i0}^T \sum_{r=1}^{k_{if}} \mu_{if}^r \mathcal{Y}_{if}^r(t_0) x_{i0} + \sum_{r=1}^{k_{if}} \mu_{if}^r \mathcal{Z}_{if}^r(t_0). \quad (28)$$

By matching the boundary condition (28), it yields the time parameter functions $\mathcal{E}_{if}^r(t_0) = 0$ and $\mathcal{T}_{if}^r(t_0) = 0$ for $1 \le r \le k_{if}$. Next, it is necessary to verify that this candidate value function satisfies (26) along the corresponding trajectories produced by the feedback gain K_{if} resulting from the minimization in (25). Or equivalently, one obtains

$$0 = \min_{K_{if} \in \overline{\mathcal{K}}_{if}} \left\{ x_{i0}^T \sum_{r=1}^{k_{if}} \mu_{if}^r \mathcal{F}_{if}^r(\varepsilon, \mathcal{Y}_{if}, K_{if}) x_{i0} + \sum_{r=1}^{k_{if}} \mu_{if}^r \mathcal{G}_{if}^r(\varepsilon, \mathcal{Y}_{if}) \right.$$
$$\left. + x_{i0}^T \sum_{r=1}^{k_{if}} \mu_{if}^r \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{E}_{if}^r(\varepsilon) x_{i0} + \sum_{r=1}^{k_{if}} \mu_{if}^r \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{T}_{if}^r(\varepsilon) \right\}. \quad (29)$$

Therefore, the derivative of the expression in (29) with respect to the admissible feedback decision gain K_{if} yields the necessary conditions for an extremum of (25) on $[t_0, t_f]$,

$$K_{if}(\varepsilon, \mathcal{Y}_{if}) = -R_{if}^{-1} B_{ii}^T \sum_{s=1}^{k_{if}} \hat{\mu}_{if}^s \mathcal{Y}_{if}^s, \qquad i = 1, \ldots, N, \tag{30}$$

where $\hat{\mu}_{if}^s \triangleq \mu_{if}^r / \mu_{if}^1$ with $\mu_{if}^1 > 0$. With the feedback decision gain (30) replaced in the expression of the bracket (29) and having $\left\{\mathcal{Y}_{if}^s\right\}_{s=1}^{k_{if}}$ evaluated on the solution trajectories (21) and (22), the time-dependent functions $\mathcal{E}_{if}^r(\varepsilon)$ and $\mathcal{T}_{if}^r(\varepsilon)$ are therefore chosen such that the sufficient condition (26) in the verification theorem is satisfied in the presence of the arbitrary value of x_{i0}; for example

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{E}_{if}^1(\varepsilon) = {} & (A_{ii} + B_{ii} K_{if}(\varepsilon))^T \mathcal{H}_{if}^1(\varepsilon) + \mathcal{H}_{if}^1(\varepsilon)(A_{ii} + B_{ii} K_{if}(\varepsilon)) \\ & + Q_{if} + K_{if}^T(\varepsilon) R_{if} K_{if}(\varepsilon)\end{aligned}$$

and for $2 \le r \le k_{if}$,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{E}_{if}^r(\varepsilon) = {} & (A_{ii} + B_{ii} K_{if}(\varepsilon))^T \mathcal{H}_{if}^r(\varepsilon) + \mathcal{H}_{if}^r(\varepsilon)(A_{ii} + B_{ii} K_{if}(\varepsilon)) \\ & + \sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[\mathcal{H}_{if}^v(\varepsilon) \Pi_{11}^f(\varepsilon) + \mathcal{H}_{if}^{k_{if}+v}(\varepsilon) \Pi_{21}^f(\varepsilon)\right] \mathcal{H}_{if}^{r-v}(\varepsilon) \\ & + \sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[\mathcal{H}_{if}^v(\varepsilon) \Pi_{12}^f(\varepsilon) + \mathcal{H}_{if}^{k_{if}+v}(\varepsilon) \Pi_{22}^f(\varepsilon)\right] \mathcal{H}_{if}^{2k_{if}+r-v}(\varepsilon)\end{aligned}$$

together with, for $1 \le r \le k_{if}$,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{T}_{if}^r(\varepsilon) = {} & \mathrm{Tr}\left\{\mathcal{H}_{if}^r(\varepsilon) \Pi_{11}^f(\varepsilon)\right\} + \mathrm{Tr}\left\{\mathcal{H}_{if}^{k_{if}+r}(\varepsilon) \Pi_{21}^f(\varepsilon)\right\} \\ & + \mathrm{Tr}\left\{\mathcal{H}_{if}^{2k_{if}+r}(\varepsilon) \Pi_{12}^f(\varepsilon)\right\} + \mathrm{Tr}\left\{\mathcal{H}_{if}^{3k_{if}+r}(\varepsilon) \Pi_{22}^f(\varepsilon)\right\}\end{aligned}$$

with the initial-value conditions $\mathcal{E}_{if}^r(t_0) = 0$ and $\mathcal{T}_{if}^r(t_0) = 0$ for $1 \le r \le k_{if}$. Therefore, the sufficient condition (26) of the verification theorem is satisfied so that the extremizing feedback decision gain (30) by decision maker i and $i = 1, \ldots, N$ becomes optimal.

Finally, the principal results of fast interactions are now summarized for linear, time-invariant stochastic systems with uncorrelated Wiener stationary distributions. For this case, the representation of performance-measure statistics has been exhibited and the risk-averse decision solutions specified.

Theorem 5 (Fast Interactions—Fast-Timescale Risk-Averse Decisions). *Consider fast interactions with the statistical control problem (9), (11), and (23) wherein* (A_{ii}, B_{ii}) *and* (A_{ii}, C_{ii}) *are stabilizable and detectable. Fix* $k_{if} \in \mathbb{N}$, *and* $\mu_{if} = \{\mu_{if}^r \geq 0\}_{r=1}^{k_{if}}$ *with* $\mu_{if}^1 > 0$. *Then, the risk-averse decision policy that minimizes the performance index (23) is exhibited in fast interactions by decision maker* i *for* $i = 1, \ldots, N$

$$u_{if}^*(t) = K_{if}^*(t)\hat{x}_{if}^*(t), \quad t \triangleq t_0 + t_f - \alpha, \quad \alpha \in [t_0, t_f]$$

$$K_{if}^*(\alpha) = -R_{if}^{-1} B_{ii}^T \sum_{r=1}^{k_{if}} \hat{\mu}_{if}^r \mathcal{H}_{if}^{r*}(\alpha), \quad \hat{\mu}_{if}^r \triangleq \frac{\mu_{if}^r}{\mu_{if}^1} \tag{31}$$

where all the parametric design freedom through $\hat{\mu}_{if}^r$ *represent different weights toward specific summary statistical performance-measures; that is, mean, variance, skewness, etc. chosen by decision maker* i *for his/her performance robustness. The optimal solutions* $\{\mathcal{H}_{if}^{r*}(\alpha)\}_{r=1}^{k_{if}}$ *satisfy the coupled time-backward matrix-valued differential equations with the terminal-value conditions* $\mathcal{H}_{if}^{1*}(t_f) = \varepsilon_i Q_{if}^f$ *and* $\mathcal{H}_{if}^{r*}(t_f) = 0$ *when* $2 \leq r \leq k_{if}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{1*}(\alpha) &= -(A_{ii} + B_{ii}K_{if}^*(\alpha))^T \mathcal{H}_{if}^{1*}(\alpha) \\ &\quad -\mathcal{H}_{if}^{1*}(\alpha)(A_{ii} + B_{ii}K_{if}^*(\alpha)) - Q_{if} - (K_{if}^*)^T(\alpha) R_{if} K_{if}^*(\alpha)\end{aligned} \tag{32}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{r*}(\alpha) &= -(A_{ii} + B_{ii}K_{if}^*(\alpha))^T \mathcal{H}_{if}^{r*}(\alpha) - \mathcal{H}_{if}^{r*}(\alpha)(A_{ii} + B_{ii}K_{if}^*(\alpha)) \\ &\quad -\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{if}^{v*}(\alpha)\Pi_{11}^f(\alpha) + \mathcal{H}_{if}^{k_{if}+v*}(\alpha)\Pi_{21}^f(\alpha)\right]\mathcal{H}_{if}^{r-v*}(\alpha) \\ &\quad -\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{if}^{v*}(\alpha)\Pi_{12}^f(\alpha) + \mathcal{H}_{if}^{k_{if}+v*}(\alpha)\Pi_{22}^f(\alpha)\right]\mathcal{H}_{if}^{2k_{if}+r-v*}(\alpha)\end{aligned} \tag{33}$$

and the optimal auxiliary solutions $\{\mathcal{H}_{if}^{k_{if}+r*}(\alpha)\}_{r=1}^{k_{if}}$ *of the time-backward differential equations with the terminal-value conditions* $\mathcal{H}_{if}^{k_{if}+1*}(t_f) = \varepsilon_i Q_{if}^f$ *and* $\mathcal{H}_{if}^{k_{if}+r*}(t_f) = 0$ *when* $2 \leq r \leq k_{if}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{k_{if}+1*}(\alpha) &= -(A_{ii} + B_{ii}K_{if}^*(\alpha))^T \mathcal{H}_{if}^{k_{if}+1*}(\alpha) - \mathcal{H}_{if}^{k_{if}+1*}(\alpha)(A_{ii} - L_{if}(\alpha)C_{ii}) \\ &\quad -\mathcal{H}_{if}^{1*}(\alpha)(L_{if}(\alpha)C_{ii}) - Q_{if}\end{aligned} \tag{34}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{k_{if}+r*}(\alpha) = &-(A_{ii} + B_{ii}K_{if}^{*}(\alpha))^{T}\mathcal{H}_{if}^{k_{if}+r*}(\alpha) - \mathcal{H}_{if}^{k_{if}+r*}(\alpha)\\ &\times(A_{ii} - L_{if}(\alpha)C_{ii})\\ &-\mathcal{H}_{if}^{r*}(\alpha)L_{if}(\alpha)C_{ii} - \sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\\ &\times\left[\mathcal{H}_{if}^{v*}(\alpha)\Pi_{11}^{f}(\alpha) + \mathcal{H}_{if}^{k_{if}+v*}(\alpha)\Pi_{21}^{f}(\alpha)\right]\mathcal{H}_{if}^{k_{if}+r-v*}(\alpha)\\ &-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{if}^{v*}(\alpha)\Pi_{12}^{f}(\alpha)\right.\\ &\qquad\left.+\mathcal{H}_{if}^{k_{if}+v*}(\alpha)\Pi_{22}^{f}(\alpha)\right]\mathcal{H}_{if}^{3k_{if}+r-v*}(\alpha) \quad (35)\end{aligned}$$

and the optimal auxiliary solutions $\{\mathcal{H}_{if}^{2k_{if}+r*}(\alpha)\}_{r=1}^{k_{if}}$ *of the time-backward differential equations with the terminal-value conditions* $\mathcal{H}_{if}^{2k_{if}+1*}(t_f) = \varepsilon_i Q_{if}^{f}$ *and* $\mathcal{H}_{if}^{2k_{if}+r*}(t_f) = 0$ *when* $2 \le r \le k_{if}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{2k_{if}+1*}(\alpha) = &-(A_{ii} - L_{if}(\alpha)C_{ii})^{T}\mathcal{H}_{if}^{2k_{if}+1*}(\alpha)\\ &-\mathcal{H}_{if}^{2k_{if}+1*}(\alpha)(A_{ii} + B_{ii}K_{if}^{*}(\alpha)) - (L_{if}(\alpha)C_{ii})^{T}\mathcal{H}_{if}^{1*}(\alpha) - Q_{if}\end{aligned} \quad (36)$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{2k_{if}+r*}(\alpha) = &-(A_{ii} - L_{if}(\alpha)C_{ii})^{T}\mathcal{H}_{if}^{2k_{if}+r*}(\alpha)\\ &-\mathcal{H}_{if}^{2k_{if}+r*}(\alpha)(A_{ii} + B_{ii}K_{if}^{*}(\alpha)) - (L_{if}(\alpha)C_{ii})^{T}\mathcal{H}_{if}^{r*}(\alpha)\\ &-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{if}^{2k_{if}+v*}(\alpha)\Pi_{11}^{f}(\alpha)\right.\\ &\qquad\left.+\mathcal{H}_{if}^{3k_{if}+v*}(\alpha)\Pi_{21}^{f}(\alpha)\right]\mathcal{H}_{if}^{r-v*}(\alpha)\\ &-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{if}^{2k_{if}+v*}(\alpha)\Pi_{12}^{f}(\alpha)\right.\\ &\qquad\left.+\mathcal{H}_{if}^{3k_{if}+v*}(\alpha)\Pi_{22}^{f}(\alpha)\right]\mathcal{H}_{if}^{2k_{if}+r-v*}(\alpha) \quad (37)\end{aligned}$$

and finally the optimal auxiliary solutions $\{\mathcal{H}_{if}^{3k_{if}+r*}(\alpha)\}_{r=1}^{k_{if}}$ *of the time-backward differential equations with the terminal-value conditions* $\mathcal{H}_{if}^{3k_{if}+1*}(t_f) = \varepsilon_i Q_{if}^f$ *and* $\mathcal{H}_{if}^{3k_{if}+r*}(t_f) = 0$ *when* $2 \le r \le k_{if}$,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{3k_{if}+1*}(\alpha) = &-(A_{ii} - L_{if}(\alpha)C_{ii})^T \mathcal{H}_{if}^{3k_{if}+1*}(\alpha)\\ &-\mathcal{H}_{if}^{3k_{if}+1*}(\alpha)(A_{ii} - L_{if}(\alpha)C_{ii})\\ &-Q_{if} - (L_{if}(\alpha)C_{ii})^T \mathcal{H}_{if}^{k_{if}+1*}(\alpha)\\ &-\mathcal{H}_{if}^{2k_{if}+1*}(\alpha)(L_{if}(\alpha)C_{ii}) \end{aligned} \tag{38}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{if}^{3k_{if}+r*}(\alpha) = &-(A_{ii} - L_{if}(\alpha)C_{ii})^T \mathcal{H}_{if}^{3k_{if}+r*}(\alpha)\\ &-\mathcal{H}_{if}^{3k_{if}+r*}(\alpha)(A_{ii} - L_{if}(\alpha)C_{ii}) - (L_{if}(\alpha)C_{ii})^T \mathcal{H}_{if}^{k_{if}+r*}(\alpha)\\ &-\mathcal{H}_{if}^{2k_{if}+r*}(\alpha)(L_{if}(\alpha)C_{ii})\\ &-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\Big[\mathcal{H}_{if}^{2k_{if}+v*}(\alpha)\Pi_{11}^f(\alpha)\\ &\qquad +\mathcal{H}_{if}^{3k_{if}+v*}(\alpha)\Pi_{21}^f(\alpha)\Big]\mathcal{H}_{if}^{k_{if}+r-v*}(\alpha)\\ &-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\Big[\mathcal{H}_{if}^{2k_{if}+v*}(\alpha)\Pi_{12}^f(\alpha)\\ &\qquad +\mathcal{H}_{if}^{3k_{if}+v*}(\alpha)\Pi_{22}^f(\alpha)\Big]\mathcal{H}_{if}^{3k_{if}+r-v*}(\alpha) \end{aligned} \tag{39}$$

where the Kalman gain $L_{if}(t) \triangleq P_{if}(t)C_{ii}^T V_{ii}^{-1}$ *is solved forwardly in time,*

$$\begin{aligned}\varepsilon_i \frac{\mathrm{d}}{\mathrm{d}t}P_{if}(t) &= P_{if}(t)A_{ii}^T + A_{ii}P_{if}(t) - P_{if}(t)C_{ii}^T V_{ii}^{-1}C_{ii}P_{if}(t) + G_i W G_i^T\\ P_{if}(t_0) &= 0\,. \end{aligned} \tag{40}$$

Remark 3. As it can be seen from (32)–(39), the calculation of the optimal feedback decision gain $K_{if}^*(\cdot)$ depends on the filter gain $L_{if}(\cdot)$ of the Kalman state estimator. Therefore, the design of optimal risk-averse decision control cannot be separated from the state estimation counterpart. In other words, the separation principle as often inherited in the LQG problem class is no longer applicable in this generalized class of stochastic control.

5 Slow Interactions

As has been alluding to, self-coordination is possible when each decision maker knows his/her place in the scheme and is prepared to carry out his/her job with the others. A useful approach for understanding the self-coordination of complex systems is to focus on slow interactions. Herein slow interactions used to map and simulate engagements within and between communities of decision makers are therefore formulated by setting $\varepsilon_i = 0$ and $\varepsilon_{ij} = 0$ of the fast timescale processes (2). Thinking about mutual influence suggests the integration of steady-state dynamics of individual process (5) with the macro-level process (1) and flows of information (3) and (4). Specifically, a typical formulation for slow interactions (with s $\sim$ "slow") is considered as follows:

$$\mathrm{d}x_{0s}(t) = \left(A_{0s}x_{0s}(t) + \sum_{i=1}^{N} B_{is}u_{is}(t) \right) \mathrm{d}t + G_{0s}\mathrm{d}w(t), \quad x_{0s}(t_0) = x_{00} \tag{41}$$

where the constant coefficients $A_{0s} \equiv A_0^i$, $B_{is} = B_{0i} - A_{0i}A_{ii}^{-1}B_{ii}$, and $G_{0s} \equiv G_0^i$.

As the slow timescale process is at work, decision maker i attempts to optimize his/her own performance. In fact, the long-term behavior or the steady-state dynamics (5) of decision maker i could yield some ill-defined terms like the integrals of the second-order statistics associated with the underlying Wiener stationary processes when substituting (5) into the utilities of decision makers, as have been well documented in [1]. However, these ill-defined terms are independent of the input decisions, $u_{is} \in U_{is} \subset L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{m_i})$. For this reason, it is expected that the optimal decision law by decision maker i obtained by solving the modified utility but assuming the only drift effect of the long-term behavior (5), $z_i(t) \triangleq -A_{ii}^{-1}(A_{i0}x_{0s}(t) + B_{ii}u_{is}(t))$ and $t \in [t_0, t_f]$, would be essentially the same as that obtained by solving the original utility except with the both diffusion and drift effects in (5). Henceforth, it requires that long-term performance, $J_{is} : \mathbb{R}^{n_0} \times U_{is} \mapsto \mathbb{R}^+$ concerning decision maker i, is measured for the impacts on slower events through the mappings

$$\begin{aligned} J_{is}(x_{00}, u_{is}) = {} & x_{0s}^T(t_f)Q_{0i}^f x_{0s}(t_f) \\ & + \int_{t_0}^{t_f} \left[x_{0s}^T(\tau)Q_{0i}x_{0s}(\tau) + z_i^T(\tau)Q_i z_i(\tau) + u_{is}^T(\tau)R_i u_{is}(\tau) \right] \mathrm{d}\tau. \end{aligned} \tag{42}$$

The constant matrices $Q_{0i}^f \in \mathbb{R}^{n_0 \times n_0}$, $Q_{0i} \in \mathbb{R}^{n_0 \times n_0}$, $Q_i \in \mathbb{R}^{n_i \times n_i}$, and $R_i \in \mathbb{R}^{m_i \times m_i}$ are real, symmetric, and positive semidefinite with R_i invertible. The relative "size" of Q_{0s}, Q_i, and R_i again enforces trade-offs between the speeds of slow and fast timescale responses and the size of the control decisions.

Next, multiperson planning must take into consideration the fact that the activities of decision makers can interfere with one another. With respect to

group interactions (41), decision maker i hence builds a model of other decision makers—their abilities, self-interest intentions, and the like—and to coordinate his/her activities around the predictions that this model makes

$$\mathrm{d}x_{is}(t) = (A_{0s}x_{is}(t) + B_{is}u_{is}(t))\mathrm{d}t + \delta_{-is}(t) + G_{0s}\mathrm{d}w(t), \quad x_{is}(t_0) = x_{00}. \quad (43)$$

Under the assumption of (A_{0s}, C_{is}) detectable, decision maker i is able to make aggregate observations $y_{is} \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{q_{i0}+q_{ii}})$ according to the relation

$$\mathrm{d}y_{is}(t) = (C_{is}x_{0s}(t) + D_{is}u_{is}(t))\,\mathrm{d}t + \mathrm{d}v_{is}(t), \quad i = 1, \ldots, N \quad (44)$$

where the aggregate observation noise $v_{is}(t)$ is an $(q_{i0}+q_{ii})$-dimensional stationary Wiener process, which is uncorrelated with $w(t)$ and has correlation of independent increments $E\left\{[v_{is}(\tau) - v_{is}(\xi)][v_{is}(\tau) - v_{is}(\xi)]^{\mathrm{T}}\right\} = V_{is}|\tau - \xi|$ with $V_{is} > 0$ for all $\tau, \xi \in [t_0, t_f]$. Moreover, all other decision makers except for decision maker i are endowed with partial knowledge about his/her observation process, in which $C_{-is} \triangleq \frac{1}{\gamma_{-is}}C_{is}$ and $D_{-is} \triangleq \frac{1}{\gamma_{-is}}D_{is}$ with scalars $\gamma_{-is} \in \mathbb{R}^+$

$$\mathrm{d}\tilde{y}_{-is}(t) = (C_{-is}x_{is}(t) + D_{-is}u_{is}(t))\,\mathrm{d}t + \mathrm{d}\eta_{-is}(t) \quad (45)$$

where the measurement noise $\eta_{-is}(t)$ is an $(q_{i0} + q_{ii})$-dimensional stationary Wiener process that correlates with neither $w(t)$ nor $v_{is}(t)$, while its correlation of independent increments $E\left\{[\eta_{-is}(\tau) - \eta_{-is}(\xi)][\eta_{-is}(\tau) - \eta_{-is}(\xi)]^{\mathrm{T}}\right\} = N_{-is}|\tau - \xi|$ with $N_{-is} > 0$ for all $\tau, \xi \in [t_0, t_f]$. As such, the perpetual signal and nominal driving term $\delta_{-is} \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{n_0})$, is generated and imposed by all neighbors around decision maker i

$$\delta_{-is}(t) = L_{-is}(t)\left[\mathrm{d}\tilde{y}_{-is}(t) - (C_{-is}\hat{x}_{is}(t) + D_{-is}u_{is}(t))\,\mathrm{d}t\right], \quad (46)$$

from which the interference intensity $L_{-is} \in C(t_0, t_f; \mathbb{R}^{n_0 \times (q_{i0}+q_{ii})})$ is yet to be defined. For greater mathematical tractability, each decision maker i with self-interest decides to retain an approximation of his/her group interactions via a model-reference estimator with filter estimates $\hat{x}_{is} \in L^2_{\mathcal{F}}(t_0, t_f; \mathbb{R}^{n_0})$ and initial values $\hat{x}_{is}(t_0) = x_{00}$

$$\mathrm{d}\hat{x}_{is}(t) = (A_{0s}\hat{x}_{is}(t) + B_{is}u_{is}(t))\,\mathrm{d}t + L_{is}(t)[\mathrm{d}y_{is}(t) - (C_{is}\hat{x}_{is}(t) + D_{is}u_{is}(t))\mathrm{d}t] \quad (47)$$

where the interaction estimate gain $L_{is} \in C(t_0, t_f; \mathbb{R}^{n_0 \times (q_{i0}+q_{ii})})$ is determined in accordance with the minimax differential game subject to the aggregate interference $L_{-is}(t)\mathrm{d}\eta_{-is}(t)$ for $t \in [t_0, t_f]$ from the group

$$\begin{aligned}\mathrm{d}\tilde{x}_{is}(t) = {} & (A_{0s} - L_{is}(t)C_{is} + L_{-is}(t)C_{-is})\,\tilde{x}_{is}(t)\mathrm{d}t \\ & + G_{0s}\mathrm{d}w(t) - L_{is}(t)\mathrm{d}v_{is}(t) + L_{-is}(t)\mathrm{d}\eta_{-is}(t), \quad \tilde{x}_{is}(t_0) = 0\,. \end{aligned} \quad (48)$$

The objective of minimax estimation is minimized by L_{is} and maximized by L_{-is} as

$$J_{is}^{e}(L_{is}, L_{-is}) = \mathrm{Tr}\left\{ M_{is}(t_f)\left[E\{\tilde{x}_{is}^{1}(t_f)(\tilde{x}_{is}^{1})^T(t_f) - \tilde{x}_{is}^{2}(t_f)(\tilde{x}_{is}^{2})^T(t_f)\}\right]\right\}$$
$$+\mathrm{Tr}\left\{ \int_{t_0}^{t_f} M_{is}(\tau)\big[E\{\tilde{x}_{is}^{1}(\tau)(\tilde{x}_{is}^{1})^T(\tau) - \tilde{x}_{is}^{2}(\tau)(\tilde{x}_{is}^{2})^T(\tau)\}\big]\,\mathrm{d}\tau\right\}$$

wherein the weighting $M_{is} \in C(t_0, t_f; \mathbb{R}^{n_0 \times n_0})$ for all the estimation errors is positive definite and the estimate errors $\tilde{x}_{is}^{1} \in L_{\mathcal{F}}^{2}(t_0, t_f; \mathbb{R}^{n_0})$ and $\tilde{x}_{is}^{2} \in L_{\mathcal{F}}^{2}(t_0, t_f; \mathbb{R}^{n_0})$ with the initial values $\tilde{x}_{is}^{1}(t_0) = 0$ and $\tilde{x}_{is}^{2}(t_0) = 0$ satisfy the stochastic differential equations

$$\mathrm{d}\tilde{x}_{is}^{1}(t) = (A_{0s} - L_{is}(t)C_{is} + L_{-is}(t)C_{-is})\,\tilde{x}_{is}^{1}(t)\mathrm{d}t + G_{0s}\mathrm{d}w(t) - L_{is}(t)\mathrm{d}v_{is}(t)$$
$$\mathrm{d}\tilde{x}_{is}^{2}(t) = (A_{0s} - L_{is}(t)C_{is} + L_{-is}(t)C_{-is})\,\tilde{x}_{is}^{2}(t)\mathrm{d}t + L_{-is}(t)\mathrm{d}\eta_{-is}(t)$$

provided the assumption of $\tilde{x}_{is}(t) \triangleq \tilde{x}_{is}^{1}(t) + \tilde{x}_{is}^{2}(t)$ with the constraint (48). As originally shown in [7], the differential game with estimation interference possesses a saddle-point equilibrium $(L_{is}^{*}, L_{-is}^{*})$ such that $J_{is}^{e}(L_{is}^{*}, L_{-is}) \le J_{is}^{e}(L_{is}^{*}, L_{-is}^{*}) \le J_{is}^{e}(L_{is}, L_{-is}^{*})$ is satisfied when decision maker i and the remaining group select their strategies $L_{is}^{*} = \underset{L_{is}}{\arg\min}\, J_{is}^{e}(L_{is}, L_{-is}^{*}) = P_{is}(t)C_{is}^{T}$ and $L_{-is}^{*} = \underset{L_{-is}}{\arg\max}\, J_{is}^{e}(L_{is}^{*}, L_{-is}) = P_{is}(t)C_{-is}^{T}$ subject to estimate-error covariances $P_{is} \in C^{1}(t_0, t_f; \mathbb{R}^{n_0 \times n_0})$ satisfying $P_{is}(t_0) = 0$

$$\frac{\mathrm{d}}{\mathrm{d}t}P_{is}(t) = A_{0s}P_{is}(t) + P_{is}(t)A_{0s}^{T} + G_{0s}WG_{0s}^{T} - P_{is}(t)(C_{is}^{T}C_{is} - C_{-is}^{T}C_{-is})P_{is}(t). \tag{49}$$

Thus far, the risk-bearing decisions of individual decision makers have been considered only in fast interactions. But it is also possible to respond to risk in slow interactions as well. Here, when it comes to decisions under uncertainty, it is not immediately evident how a ranking of consequences leads to an ordering of actions, since each action will simply imply a chi-squared probabilistic mix of performance whose description (42) is now rewritten conveniently for the sequel analysis

$$J_{is}(x_{00}; u_{is}) = x_{is}^{T}(t_f)Q_{0is}^{f}x_{is}(t_f)$$
$$+\int_{t_0}^{t_f}\big[x_{is}^{T}(\tau)Q_{0is}x_{is}(\tau) + 2x_{is}^{T}(\tau)Q_{is}u_{is}(\tau)$$
$$+u_{is}^{T}(\tau)R_{is}u_{is}(\tau)\big]\,\mathrm{d}\tau \tag{50}$$

wherein the constant weighting matrices $Q_{0is} \triangleq Q_{0i} + (A_{ii}^{-1}A_{i0})^T Q_i (A_{ii}^{-1}A_{i0})$, $Q_{is} \triangleq (A_{ii}^{-1}A_{i0}) Q_i (A_{ii}^{-1}B_{ii})$, $R_{is} \triangleq R_i + (A_{ii}^{-1}B_{ii})^T Q_i (A_{ii}^{-1}B_{ii})$, and $Q_{0is}^f \triangleq Q_{0i}^f$.

Having been dissatisfied with the perceived level of utility risk, individual decision maker decides to construct his/her action repertoire. The objective for each decision maker is the reliable attainment of his/her own utility and preferences (50) by choosing appropriate decision strategies for the underlying linear dynamical system (43) and its approximation (47) and (48). The noncooperative aspect δ_{-is} governed by (46) implies that the other decision makers have been assumed not to collaborate in trying to attain this goal reliably for decision maker i. Depending on the information η_{is} and the set of strategies Γ_{is} the decision makers like to choose from, the actions of the decision makers are then determined by the relations; that is, $\gamma_{is} : \Gamma_{is} \mapsto U_{is}$ and $u_{is} = \gamma_{is}(\eta_{is})$. Henceforth, the performance value of (50) and its robustness depend on the information η_{is} that decision maker i has for interactions and his/her strategy space. Furthermore, the performance distribution of (50) obviously also depends for each decision maker i on the pursued actions δ_{-is} of the other decision makers.

With interests of mutual modeling and self-direction, each decision maker no longer needs prior knowledge of the remaining decision makers' decisions and thus cannot be certain of how the other decision makers select their pursued actions. It is reasonable to assume that decision maker i may instead choose to optimize his/her decision and performance against the worst possible set of decision strategies, which the other decision makers could choose. Henceforth, it is assumed that decision makers are constrained to use minimax-state estimates $\hat{x}_{is}(t)$ for their responsive decision implementation. Due to the fact that the interaction model (47) and (48) is linear and the path-wise performance-measure (50) is quadratic, the information structure for optimal decisions is now considered to be linear. Therefore, it is reasonable to restrict the search for the optimal decision laws to linear time-varying decision feedback laws generated from the minimax-state estimates $\hat{x}_{is}(t)$. That is, $\eta_{is} \triangleq (t, \hat{x}_{is}(t))$ and $\Gamma_{is} \triangleq \{u_{is}(t) = \gamma_{is}(t, \hat{x}_{is}(t)) \text{ and } u_{is} \in U_{is}\}$ for $i = 1, \ldots, N$. In view of the common knowledge (47) and state-decision coupling utility (50), it is reasonable to construct probing decisions that can bring to bear additional information about expected performance and its certainty according to the relation

$$u_{is}(t) \triangleq K_{is}(t)\hat{x}_{is}(t) + p_{is}(t)\,, \qquad i = 1, \ldots, N \tag{51}$$

wherein the admissible slow timescale decision gain $K_{is} \in C(t_0, t_f; \mathbb{R}^{m_i \times n_0})$ and affine slow timescale correction $p_{is} \in C(t_0, t_f; \mathbb{R}^{m_i})$ are to be determined in some appropriate sense.

What next are the aggregate interactions (47) and (48) by decision maker i with self-interest, which come from the implementation of action (51)

$$\mathrm{d}z_{is}(t) = (F_{is}(t)z_{is}(t) + l_{is}(t))\mathrm{d}t + G_{is}(t)\mathrm{d}w_{is}(t)\,, \quad z_{is}(t_0) = z_{is}^0 \tag{52}$$

where the aggregate system states, parameters, and process disturbances are given by

$$z_{is} \triangleq \begin{bmatrix} \hat{x}_{is} \\ \tilde{x}_{is} \end{bmatrix}, \; z_{is}(t_0) \triangleq \begin{bmatrix} x_{00} \\ 0 \end{bmatrix}, \; l_{is} \begin{bmatrix} B_{is} p_{is} \\ 0 \end{bmatrix}, \; w_{is} \triangleq \begin{bmatrix} w \\ v_{is} \\ \eta_{-is} \end{bmatrix}, \; W_{is} \triangleq \begin{bmatrix} W & 0 & 0 \\ 0 & V_{is} & 0 \\ 0 & 0 & N_{-is} \end{bmatrix}$$

$$F_{is} \triangleq \begin{bmatrix} A_{0s} + B_{is} K_{is} & L_{is}^* C_{is} \\ 0 & A_{0s} - L_{is}^* C_{is} + L_{-is}^* C_{-is} \end{bmatrix}, \quad G_{is} \triangleq \begin{bmatrix} 0 & L_{is}^* & 0 \\ G_{0s} & -L_{is}^* & L_{-is}^* \end{bmatrix}$$

$$E\{[w_{is}(\tau) - w_{is}(\xi)][w_{is}(\tau) - w_{is}(\xi)]^{\mathrm{T}}\} = W_{is}|\tau - \xi|, \qquad \forall \tau, \xi \in [t_0, t_f].$$

Then, for given admissible affine p_{is} and feedback decision K_{is}, the performance-measure (42) is seen as the "cost-to-go," $J_{is}(\alpha, z_{is}^{\alpha})$ when parameterizing the initial condition (t_0, z_{is}^0) to any arbitrary pair $(\alpha, z_{is}^{\alpha})$

$$\begin{aligned} J_{is}(\alpha, z_{is}^{\alpha}) = {} & z_{is}^T(t_f) O_{is}^f z_{is}(t_f) \\ & + \int_{\alpha}^{t_f} \left[z_{is}^T(\tau) O_{is}(\tau) z_{is}(\tau) + 2 z_{is}^T(\tau) N_{is}(\tau) \right. \\ & \left. + p_{is}^T(\tau) R_{is} p_{is}(\tau) \right] \mathrm{d}\tau \end{aligned} \tag{53}$$

wherein

$$N_{is} \triangleq \begin{bmatrix} K_{is}^T R_{is} p_{is} + Q_{is} p_{is} \\ Q_{is} p_{is} \end{bmatrix}, \qquad O_{is}^f \triangleq \begin{bmatrix} Q_{i0}^f & Q_{i0}^f \\ Q_{i0}^f & Q_{i0}^f \end{bmatrix}$$

$$O_{is} \triangleq \begin{bmatrix} Q_{0is} + K_{is}^T R_{is} K_{is} + 2 Q_{is} K_{is} & Q_{0is} \\ Q_{0is} + 2 Q_{is} K_{is} & Q_{0is} \end{bmatrix}.$$

So far there are two types of information, i.e., *process information* (52) and *goal information* (53) have been given in advance to the control decision policy (51). Since there is the external disturbance $w_{is}(\cdot)$ affecting the closed-loop performance, the control decision policy now needs additional information about performance variations. This is *coupling information* and thus also known as *performance information*. The questions of how to characterize and influence performance information are then answered by adaptive cumulants (aka semi-invariants) associated with the performance-measure (53) in details below.

Associated with each decision maker i, the first and second characteristic functions or the moment and cumulant-generating functions of (53) are defined by

$$\varphi_{is}\left(\alpha, z_{is}^{\alpha}; \theta_{is}\right) \triangleq E\left\{\exp\left(\theta_{is} J_{is}\left(\alpha, z_{is}^{\alpha}\right)\right)\right\} \tag{54}$$

$$\psi_{is}\left(\alpha, z_{is}^{\alpha}; \theta_{is}\right) \triangleq \ln\left\{\varphi_{is}\left(\alpha, z_{is}^{\alpha}; \theta_{is}\right)\right\} \tag{55}$$

for some small parameters θ_{is} in an open interval about 0 while $\ln\{\cdot\}$ denotes the natural logarithmic transformation of the first characteristic function.

Theorem 6 (Slow Interactions—cumulant-Generating Function). *Let θ_{is} be a small positive parameter and $\alpha \in [t_0, t_f]$ be a running variable. Further let*

$$\varphi_{is}\left(\alpha, z_{is}^{\alpha}; \theta_{is}\right) = \varrho_{is}(\alpha; \theta_{is}) \exp\{(z_{is}^{\alpha})^T \Upsilon_{is}(\alpha; \theta_{is}) z_{is}^{\alpha} + 2(z_{is}^{\alpha})^T \eta_{is}(\alpha; \theta_{is})\} \quad (56)$$

$$\upsilon_{is}(\alpha; \theta_{is}) = \ln\{\varrho_{is}(\alpha; \theta_{is})\}, \quad i = 1, \ldots, N \quad (57)$$

Under the assumption of (A_{0s}, B_{is}) and (A_{0s}, C_{is}) stabilizable and detectable, the cumulant-generating function that compactly and robustly represents the uncertainty of performance distribution (53) is given by

$$\psi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is}) = (z_{is}^{\alpha})^T \Upsilon_{is}(\alpha; \theta_{is}) z_{is}^{\alpha} + 2(z_{is}^{\alpha})^T \eta_{is}(\alpha; \theta_{is}) + \upsilon_{is}(\alpha; \theta_{is}) \quad (58)$$

subject to

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\alpha}\Upsilon_{is}(\alpha; \theta_{is}) &= -F_{is}^T(\alpha)\Upsilon_{is}(\alpha; \theta_{is}) - \Upsilon_{is}(\alpha; \theta_{is})F_{is}(\alpha) - \theta_{is}O_{is}(\alpha) \\ &\quad - 2\Upsilon_{is}(\alpha; \theta_{is})G_{is}(\alpha)W_{is}G_{is}^T(\alpha)\Upsilon_{is}(\alpha; \theta_{is}), \quad \Upsilon_{is}(t_f; \theta_{is}) = \theta_{is}O_{is}^f \end{aligned} \quad (59)$$

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\alpha}\eta_{is}(\alpha; \theta_{is}) &= -F_{is}^T(\alpha)\eta_{is}(\alpha; \theta_{is}) - \Upsilon_{is}(\alpha; \theta_{is})l_{is}(\alpha) - \theta_{is}N_{is}(\alpha) \\ \eta_{is}(t_f; \theta_{is}) &= 0 \end{aligned} \quad (60)$$

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\alpha}\upsilon_{is}(\alpha; \theta_{is}) &= -\mathrm{Tr}\{\Upsilon_{is}(\alpha; \theta_{is})G_{is}(\alpha)W_{is}G_{is}^T(\alpha)\} - 2\eta_{is}^T(\alpha; \theta_{is})l_{is}(\alpha) \\ &\quad - \theta_{is}p_{is}^T(\alpha)R_{is}p_{is}(\alpha), \qquad \upsilon_{is}(t_f; \theta_{is}) = 0. \end{aligned} \quad (61)$$

Proof. For shorthand notations, it is convenient to let the first characteristic function denoted by $\varpi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is}) \triangleq \exp\{\theta_{is}J_{is}(\alpha, z_{is}^{\alpha})\}$. The moment-generating function becomes $\varphi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is}) = E\{\varpi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is})\}$ with time derivative of

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\alpha}\varphi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is}) \\ &\quad = -\theta_{is}\left[(z_{is}^{\alpha})^T O_{is}(\alpha)z_{is}^{\alpha} + 2(z_{is}^{\alpha})^T N_{is}(\alpha) + p_{is}^T(\alpha)R_{is}p_{is}(\alpha)\right]\varphi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is}). \end{aligned}$$

Using the standard Ito's formula, one gets

$$\begin{aligned} \mathrm{d}\varphi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is}) &= E\{\mathrm{d}\varpi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is})\} \\ &= \frac{\partial}{\partial\alpha}\varphi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is})\mathrm{d}\alpha + \frac{\partial}{\partial z_{is}^{\alpha}}\varphi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is})\left[F_{is}(\alpha)z_{is}^{\alpha} + l_{is}(\alpha)\right]\mathrm{d}\alpha \\ &\quad + \frac{1}{2}\mathrm{Tr}\left\{\frac{\partial^2}{\partial(z_{is}^{\alpha})^2}\varphi_{is}(\alpha, z_{is}^{\alpha}; \theta_{is})G_{is}(\alpha)W_{is}G_{is}^T(\alpha)\right\}\mathrm{d}\alpha \end{aligned}$$

when combined with (56) leads to

$$
\begin{aligned}
&-\theta_{is}\left[(z_{is}^{\alpha})^{T} O_{is}(\alpha) z_{is}^{\alpha}+2(z_{is}^{\alpha})^{T} N_{is}(\alpha)+p_{is}^{T}(\alpha) R_{is} p_{is}(\alpha)\right] \varphi_{is}(\alpha, z_{is}^{\alpha} ; \theta_{is}) \\
&\quad=\left\{\frac{\frac{\mathrm{d}}{\mathrm{d}\alpha} \varrho_{is}(\alpha ; \theta_{is})}{\varrho_{is}(\alpha ; \theta_{is})}+(z_{is}^{\alpha})^{T} \frac{\mathrm{d}}{\mathrm{d}\alpha} \Upsilon_{is}(\alpha ; \theta_{is}) z_{is}^{\alpha}+2(z_{is}^{\alpha})^{T} \frac{\mathrm{d}}{\mathrm{d}\alpha} \eta_{is}(\alpha ; \theta_{is})\right. \\
&\qquad+(z_{is}^{\alpha})^{T} \Upsilon_{is}(\alpha ; \theta_{is}) F_{is}(\alpha) z_{is}^{\alpha}+(z_{is}^{\alpha})^{T} F_{is}^{T}(\alpha) \Upsilon_{is}(\alpha ; \theta_{is}) z_{is}^{\alpha} \\
&\qquad+2(z_{is}^{\alpha})^{T} \Upsilon_{is}(\alpha ; \theta_{is}) l_{is}(\alpha)+2(z_{is}^{\alpha})^{T} F_{is}^{T}(\alpha) \eta_{is}(\alpha ; \theta_{is}) \\
&\qquad+2 \eta_{is}^{T}(\alpha ; \theta_{is}) l_{is}(\alpha)+\operatorname{Tr}\left\{\Upsilon_{is}(\alpha ; \theta_{is}) G_{is}(\alpha) W_{is} G_{is}^{T}(\alpha)\right\} \\
&\qquad\left.+2(z_{is}^{\alpha})^{T} \Upsilon_{is}(\alpha ; \theta_{is}) G_{is}(\alpha) W_{is} G_{is}^{T}(\alpha) \Upsilon_{is}(\alpha ; \theta_{is}) z_{is}^{\alpha}\right\} \varphi_{is}(\alpha, z_{is}^{\alpha} ; \theta_{is}). \quad (62)
\end{aligned}
$$

To have all terms in (62) to be independent of arbitrary z_{is}^{α}, it requires the matrix, vector, and scalar-valued differential equations (59)–(61) with the terminal-value conditions hold true. □

By definition, the mathematical statistics associated with (53) that provide performance information for the decision process taken by decision maker i can best be generated by the MacLaurin series expansion of the cumulant-generating function (58)

$$
\begin{aligned}
\psi_{is}(\alpha, z_{is}^{\alpha} ; \theta_{is}) &\triangleq \sum_{k_{is}=1}^{\infty} \kappa_{is}^{k_{is}} \frac{(\theta_{is})^{k_{is}}}{k_{is}!} \\
&=\sum_{k_{is}=1}^{\infty} \frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}} \psi_{is}(\alpha, z_{is}^{\alpha} ; \theta_{is})\Big|_{\theta_{is}=0} \frac{(\theta_{is})^{k_{is}}}{k_{is}!} \quad (63)
\end{aligned}
$$

in which $\kappa_{is}^{k_{is}}$'s are called the performance-measure statistics associated with decision maker i for $i = 1, \ldots, N$. Notice that the series coefficients in (63) are identified as

$$
\begin{aligned}
\frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}} \psi_{is}(\alpha, z_{is}^{\alpha} ; \theta_{is})\Big|_{\theta_{is}=0} &= (z_{is}^{\alpha})^{T} \frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}} \Upsilon_{is}(\alpha ; \theta_{is})\Big|_{\theta_{is}=0} z_{is}^{\alpha} \\
&\quad+2(z_{is}^{\alpha})^{T} \frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}} \eta_{is}(\alpha ; \theta_{is})\Big|_{\theta_{is}=0} \\
&\quad+\frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}} \upsilon_{is}(\alpha ; \theta_{is})\Big|_{\theta_{is}=0}. \quad (64)
\end{aligned}
$$

For notational convenience, the necessary definitions are introduced as follows:

$$H_{is}(\alpha,k_{is}) \triangleq \frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}}\Upsilon_{is}(\alpha;\theta_{is})\bigg|_{\theta_{is}=0}, \quad \breve{D}_{is}(\alpha,k_{is}) \triangleq \frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}}\eta_{is}(\alpha;\theta_{is})\bigg|_{\theta_{is}=0}$$

$$D_{is}(\alpha,k_{is}) \triangleq \frac{\partial^{(k_{is})}}{\partial(\theta_{is})^{(k_{is})}}\upsilon_{is}(\alpha;\theta_{is})\bigg|_{\theta_{is}=0}$$

which leads to

$$\kappa_{is}^{k_{is}} = (z_{is}^{\alpha})^T H_{is}(\alpha,k_{is})z_{is}^{\alpha} + 2(z_{is}^{\alpha})^T \breve{D}_{is}(\alpha,k_{is}) + D_{is}(\alpha,k_{is}).$$

The result below contains a tractable method of generating performance-measure statistics that provides measures of the amount, value, and the design of performance information structures in time domain. This computational procedure is preferred to that of (64) for the reason that the cumulant-generating equations (59)–(61) now allow the incorporation of classes of linear feedback strategies in the statistical control problems.

Theorem 7 (Slow Interactions—Performance-Measure Statistics). *Assume interaction dynamics by decision maker i and $i = 1,\ldots,N$ is described by (52) and (53) in which the pairs (A_{0s}, B_{is}) and (A_{0s}, C_{is}) are stabilizable and detectable. For $k_{is} \in \mathbb{N}$ fixed, the k_{is}th statistics of performance-measure (53) is given by*

$$\kappa_{is}^{k_{is}} = (z_{is}^{\alpha})^T H_{is}(\alpha,k_{is})z_{is}^{\alpha} + 2(z_{is}^{\alpha})^T \breve{D}_{is}(\alpha,k_{is}) + D_{is}(\alpha,k_{is}). \tag{65}$$

where the cumulant-generating solutions $\{H_{is}(\alpha,r)\}_{r=1}^{k_{is}}$, $\{\breve{D}_{is}(\alpha,r)\}_{r=1}^{k_{is}}$, and $\{D_{is}(\alpha,r)\}_{r=1}^{k_{is}}$ evaluated at $\alpha = t_0$ satisfy the time-backward matrix differential equations (with the dependence upon $K_{is}(\alpha)$ and $p_{is}(\alpha)$ suppressed)

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}H_{is}(\alpha,1) = -F_{is}^T H_{is}(\alpha,1) - H_{is}(\alpha,1)F_{is}(\alpha) - O_{is}(\alpha) \tag{66}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}H_{is}(\alpha,r) &= -F_{is}^T H_{is}(\alpha,r) - H_{is}(\alpha,r)F_{is}(\alpha)\\ &\quad - \sum_{s=1}^{r-1}\frac{2r!}{s!(r-s)!}H_{is}(\alpha,s)G_{is}(\alpha)W_{is}G_{is}^T(\alpha)H_{is}(\alpha,r-s)\,,\\ &\quad \times 2 \le r \le k_{is}\end{aligned} \tag{67}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\breve{D}_{is}(\alpha,1) = -F_{is}^T(\alpha)\breve{D}_{is}(\alpha,1) - H_{is}(\alpha,1)l_{is}(\alpha) - N_{is}(\alpha) \tag{68}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\breve{D}_{is}(\alpha,r) = -F_{is}^T(\alpha)\breve{D}_{is}(\alpha,r) - H_{is}(\alpha,r)l_{is}(\alpha)\,, \quad 2 \le r \le k_{is} \tag{69}$$

and finally

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} D_{is}(\alpha, 1) = -\mathrm{Tr}\left\{H_{is}(\alpha, 1) G_{is}(\alpha) W_{is} G_{is}^T(\alpha)\right\} - 2(\breve{D}_{is})^T(\alpha, 1) l_{is}(\alpha) - p_{is}^T(\alpha) R_{is} p_{is}(\alpha) \tag{70}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} D_{is}(\alpha, r) = -\mathrm{Tr}\left\{H_{is}(\alpha, r) G_{is}(\alpha) W_{is} G_{is}^T(\alpha)\right\} - 2(\breve{D}_{is})^T(\alpha, r) l_{is}(\alpha), \qquad 2 \le r \le k_{is} \tag{71}$$

where the terminal-value conditions are $H_{is}(t_f, 1) = O_{is}^f$, $H_{is}(t_f, r) = 0$ *for* $2 \le r \le k_{is}$, $\breve{D}_{is}(t_f, r) = 0$ *for* $1 \le r \le k_{is}$, *and* $D_{is}(t_f, r) = 0$ *for* $1 \le r \le k_{is}$.

Proof. The expression of performance-measure statistics (65) is readily justified by using the result (64). What remains is to show that the solutions $H_{is}(\alpha, r)$, $\breve{D}_{is}(\alpha, r)$, and $D_{is}(\alpha, r)$ for $1 \le r \le k_{is}$ indeed satisfy the dynamical equations (66)–(71). In fact the equations (66)–(71), which are satisfied by the solutions $H_{is}(\alpha, r)$, $\breve{D}_{is}(\alpha, r)$, and $D_{is}(\alpha, r)$ can be obtained by repeatedly taking time derivatives with respect to θ_{is} of the supporting equations (59)–(61) together with the assumption of (A_{0s}, B_{is}) and (A_{0s}, C_{is}) stabilizable and detectable on $[t_0, t_f]$. □

Remark 4. It is worth the time to observe that this research investigation focuses on the class of optimal statistical control problems whose performance index reflects the intrinsic performance variability introduced by process noise stochasticity. It should also not be forgotten that all the performance-measure statistics (65) depend in part on the initial condition $z_{is}(\alpha)$. Although different states $z_{is}(t)$ and $t \in [\alpha, t_f]$ will result in different values for the "performance-to-come" (53), the performance-measure statistics are, however, the functions of time-backward evolutions of the cumulant-generating solutions $H_{is}(\alpha, r)$, $\breve{D}_{is}(\alpha, r)$, and $D_{is}(\alpha, r)$ that totally ignore all the intermediate values $z_{is}(t)$. This fact therefore makes the new optimization problem as being considered in optimal statistical control particularly unique, as compared with the more traditional dynamic programming class of investigations. In other words, the time-backward trajectories (66)–(71) should be considered as the "new" dynamical equations for the optimal statistical control, from which the resulting Mayer optimization [5] and associated value function in the framework of dynamic programming therefore depend on these "new" states $H_{is}(\alpha, r)$, $\breve{D}_{is}(\alpha, r)$, and $D_{is}(\alpha, r)$; not the classical states $z_{is}(t)$ as in the traditional school of thinking.

In the design of a decision process in which the information process about performance variations is embedded with K_{is} and p_{is}, it is convenient to rewrite the results (66)–(71) in accordance of the following matrix and vector partitions:

$$H_{is}(\cdot, r) = \begin{bmatrix} H_{is}^{11}(\cdot, r) & H_{is}^{12}(\cdot, r) \\ H_{is}^{21}(\cdot, r) & H_{is}^{22}(\cdot, r) \end{bmatrix}, \quad \breve{D}_{is}(\cdot, r) = \begin{bmatrix} \breve{D}_{is}^{11}(\cdot, r) \\ \breve{D}_{is}^{21}(\cdot, r) \end{bmatrix}$$

$$G_{is}(\cdot) W_{is} G_{is}^T(\cdot) = \begin{bmatrix} \Pi_{11}^s(\cdot) & \Pi_{12}^s(\cdot) \\ \Pi_{21}^s(\cdot) & \Pi_{22}^s(\cdot) \end{bmatrix}$$

provided that the shorthand notations $\Pi_{11}^s = L_{is}^* V_{is}(L_{is}^*)^T$, $\Pi_{12}^s = \Pi_{21}^s = -\Pi_{11}^s$, and $\Pi_{22}^s = G_{0s}WG_{0s}^T + L_{is}^* V_{is}(L_{is}^*)^T + L_{-is}^* N_{-is}(L_{-is}^*)^T$; wherein the second-order statistics associated with the $(q_{0i} + q_{ii})$-dimensional stationary Wiener process v_{is} is given by

$$V_{is} = \begin{bmatrix} V_{0i} + \varepsilon_i (C_i A_{ii}^{-1} G_i) W (C_i A_{ii}^{-1} G_i)^T & \sqrt{\varepsilon_i} (C_i A_{ii}^{-1} G_i) W (C_{ii} A_{ii}^{-1} G_i)^T \\ 0 & V_{ii} + (C_{ii} A_{ii}^{-1} G_i) W (C_{ii} A_{ii}^{-1} G_i)^T \end{bmatrix}.$$

For notational simplicity, k_{is}-tuple variables $\mathcal{H}_{is}^{11}(\cdot)$, $\mathcal{H}_{is}^{12}(\cdot)$, $\mathcal{H}_{is}^{21}(\cdot)$, $\mathcal{H}_{is}^{22}(\cdot)$, $\check{\mathcal{D}}_{is}^{11}(\cdot)$, $\check{\mathcal{D}}_{is}^{21}(\cdot)$, and $\mathcal{D}_{is}(\cdot)$ are introduced as the new dynamical states for decision maker i

$$\begin{aligned}
\mathcal{H}_{is}^{11}(\cdot) &\triangleq (\mathcal{H}_{is,1}^{11}(\cdot), \ldots, \mathcal{H}_{is,k_{is}}^{11}(\cdot)) \equiv (H_{is}^{11}(\cdot,1), \ldots, H_{is}^{11}(\cdot,k_{is})) \\
\mathcal{H}_{is}^{12}(\cdot) &\triangleq (\mathcal{H}_{is,1}^{12}(\cdot), \ldots, \mathcal{H}_{is,k_{is}}^{12}(\cdot)) \equiv (H_{is}^{12}(\cdot,1), \ldots, H_{is}^{12}(\cdot,k_{is})) \\
\mathcal{H}_{is}^{21}(\cdot) &\triangleq (\mathcal{H}_{is,1}^{21}(\cdot), \ldots, \mathcal{H}_{is,k_{is}}^{21}(\cdot)) \equiv (H_{is}^{21}(\cdot,1), \ldots, H_{is}^{21}(\cdot,k_{is})) \\
\mathcal{H}_{is}^{22}(\cdot) &\triangleq (\mathcal{H}_{is,1}^{22}(\cdot), \ldots, \mathcal{H}_{is,k_{is}}^{22}(\cdot)) \equiv (H_{is}^{22}(\cdot,1), \ldots, H_{is}^{22}(\cdot,k_{is})) \\
\check{\mathcal{D}}_{is}^{11}(\cdot) &\triangleq (\check{\mathcal{D}}_{is,1}^{11}(\cdot), \ldots, \check{\mathcal{D}}_{is,k_{is}}^{11}(\cdot)) \equiv (\check{D}_{is}^{11}(\cdot,1), \ldots, \check{D}_{is}^{11}(\cdot,k_{is})) \\
\check{\mathcal{D}}_{is}^{21}(\cdot) &\triangleq (\check{\mathcal{D}}_{is,1}^{21}(\cdot), \ldots, \check{\mathcal{D}}_{is,k_{is}}^{21}(\cdot)) \equiv (\check{D}_{is}^{21}(\cdot,1), \ldots, \check{D}_{is}^{21}(\cdot,k_{is})) \\
\mathcal{D}_{is}(\cdot) &\triangleq (\mathcal{D}_{is,1}(\cdot), \ldots, \mathcal{D}_{is,k_{is}}(\cdot)) \equiv (D_{is}(\cdot,1), \ldots, D_{is}(\cdot,k_{is}))
\end{aligned}$$

which are satisfying the matrix, vector, and scalar-valued differential equations (66)–(71). Furthermore, the right members of the matrix, vector, and scalar-valued differential equations (66)–(71) are considered as the mappings on the $[t_0, t_f]$ with the rules of action

$$\begin{aligned}
&F_{is,1}^{11}(\alpha, H_{is}^{11}(\alpha), H_{is}^{12}(\alpha), H_{is}^{21}(\alpha), K_{is}(\alpha)) \triangleq -(A_{0s} + B_{is}K_{is}(\alpha))^T H_{is}^{11}(\alpha,1) \\
&\quad - H_{is}^{11}(\alpha,1)(A_{0s} + B_{is}K_{is}(\alpha)) - Q_{0is} - K_{is}^T(\alpha) R_{is} K_{is}(\alpha) - 2Q_{is}K_{is}(\alpha)
\end{aligned} \tag{72}$$

when $2 \le r \le k_{is}$

$$\begin{aligned}
&F_{is,r}^{11}(\alpha, H_{is}^{11}(\alpha), H_{is}^{12}(\alpha), H_{is}^{21}(\alpha), K_{is}(\alpha)) \\
&\quad \triangleq -(A_{0s} + B_{is}K_{is}(\alpha))^T H_{is}^{11}(\alpha,r) - H_{is}^{11}(\alpha,r)(A_{0s} + B_{is}K_{is}(\alpha)) \\
&\quad \sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[H_{is}^{11}(\alpha,v)\Pi_{11}^s(\alpha) + H_{is}^{12}(\alpha,v)\Pi_{21}^s(\alpha) \right] H_{is}^{11}(\alpha, r-v) \\
&\quad - \sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[H_{is}^{11}(\alpha,v)\Pi_{12}^s(\alpha) + H_{is}^{12}(\alpha,v)\Pi_{22}^s(\alpha) \right] H_{is}^{21}(\alpha, r-v)
\end{aligned} \tag{73}$$

$$
\begin{aligned}
&F_{is,1}^{12}(\alpha, H_{is}^{11}(\alpha), H_{is}^{12}(\alpha), H_{is}^{22}(\alpha), K_{is}(\alpha)) \\
&\quad \triangleq -(A_{0s} + B_{is}K_{is}(\alpha))^T H_{is}^{12}(\alpha, 1) - H_{is}^{11}(\alpha, 1)(L_{is}^*(\alpha)C_{is}) \\
&\qquad -H_{is}^{12}(\alpha, 1)(A_{0s} - L_{is}^*(\alpha)C_{is} + L_{-is}^*(\alpha)C_{-is}) - Q_{0is}
\end{aligned} \tag{74}
$$

when $2 \leq r \leq k_{is}$

$$
\begin{aligned}
&F_{is,r}^{12}(\alpha, H_{is}^{11}(\alpha), H_{is}^{12}(\alpha), H_{is}^{22}(\alpha), K_{is}(\alpha)) \triangleq -(A_{0s} + B_{is}K_{is}(\alpha))^T H_{is}^{12}(\alpha, r) \\
&-H_{is}^{11}(\alpha, r)(L_{is}^*(\alpha)C_{is}) - H_{is}^{12}(\alpha, r)(A_{0s} - L_{is}^*(\alpha)C_{is} + L_{-is}^*(\alpha)C_{-is}) \\
&-\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[H_{is}^{11}(\alpha, v)\Pi_{11}^s(\alpha) + H_{is}^{12}(\alpha, v)\Pi_{21}^s(\alpha)\right] H_{is}^{12}(\alpha, r-v) \\
&-\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[H_{is}^{11}(\alpha, v)\Pi_{12}^s(\alpha) + H_{is}^{12}(\alpha, v)\Pi_{22}^s(\alpha)\right] H_{is}^{22}(\alpha, r-v)
\end{aligned} \tag{75}
$$

$$
\begin{aligned}
&F_{is,1}^{21}(\alpha, H_{is}^{11}(\alpha), H_{is}^{21}(\alpha), H_{is}^{22}(\alpha), K_{is}(\alpha)) \triangleq -Q_{0is} - 2Q_{is}K_{is}(\alpha) \\
&\quad -(A_{0s} - L_{is}^*(\alpha)C_{is} + L_{-is}^*(\alpha)C_{-is})^T H_{is}^{21}(\alpha, 1) \\
&\quad -H_{is}^{21}(\alpha, 1)(A_{0s} + B_{is}K_{is}(\alpha)) - (L_{is}^*(\alpha)C_{is})^T H_{is}^{11}(\alpha, 1)
\end{aligned} \tag{76}
$$

when $2 \leq r \leq k_{is}$

$$
\begin{aligned}
&F_{is,1}^{21}(\alpha, H_{is}^{11}(\alpha), H_{is}^{21}(\alpha), H_{is}^{22}(\alpha), K_{is}(\alpha)) \triangleq -H_{is}^{21}(\alpha, r)(A_{0s} + B_{is}K_{is}(\alpha)) \\
&\quad -(A_{0s} - L_{is}^*(\alpha)C_{is} + L_{-is}^*(\alpha)C_{-is})^T H_{is}^{21}(\alpha, r) - (L_{is}^*(\alpha)C_{is})^T H_{is}^{11}(\alpha, r) \\
&\quad -\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[H_{is}^{21}(\alpha, v)\Pi_{11}^s(\alpha) + H_{is}^{22}(\alpha, v)\Pi_{21}^s(\alpha)\right] H_{is}^{11}(\alpha, r-v) \\
&\quad -\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!} \left[H_{is}^{21}(\alpha, v)\Pi_{12}^s(\alpha) + H_{is}^{22}(\alpha, v)\Pi_{22}^s(\alpha)\right] H_{is}^{21}(\alpha, r-v)
\end{aligned} \tag{77}
$$

$$
\begin{aligned}
&F_{is,1}^{22}(\alpha, H_{is}^{12}(\alpha), H_{is}^{21}(\alpha), H_{is}^{22}(\alpha)) \triangleq -(L_{is}^*(\alpha)C_{is})^T H_{is}^{12}(\alpha, 1) \\
&\quad -(A_{0s} - L_{is}^*(\alpha)C_{is} + L_{-is}^*(\alpha)C_{-is})^T H_{is}^{22}(\alpha, 1) - Q_{0is} \\
&\quad -H_{is}^{22}(\alpha, 1)(A_{0s} - L_{is}^*(\alpha)C_{is} + L_{-is}^*(\alpha)C_{-is}) - H_{is}^{21}(\alpha, 1)(L_{is}^*(\alpha)C_{is})
\end{aligned} \tag{78}
$$

when $2 \leq r \leq k_{is}$

$$
\begin{aligned}
&F_{is,r}^{22}(\alpha, H_{is}^{12}(\alpha), H_{is}^{21}(\alpha), H_{is}^{22}(\alpha)) \triangleq -(L_{is}^*(\alpha)C_{is})^T H_{is}^{12}(\alpha, r) \\
&\quad -(A_{0s} - L_{is}^*(\alpha)C_{is} + L_{-is}^*(\alpha)C_{-is})^T H_{is}^{22}(\alpha, r)
\end{aligned}
$$

$$-H_{is}^{22}(\alpha,r)(A_{0s}-L_{is}^{*}(\alpha)C_{is}+L_{-is}^{*}(\alpha)C_{-is})-H_{is}^{21}(\alpha,r)(L_{is}^{*}(\alpha)C_{is})$$
$$-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[H_{is}^{21}(\alpha,v)\Pi_{11}^{s}(\alpha)+H_{is}^{22}(\alpha,v)\Pi_{21}^{s}(\alpha)\right]H_{is}^{12}(\alpha,r-v)$$
$$-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[H_{is}^{21}(\alpha,v)\Pi_{12}^{s}(\alpha)+H_{is}^{22}(\alpha,v)\Pi_{22}^{s}(\alpha)\right]H_{is}^{22}(\alpha,r-v) \quad (79)$$

$$\breve{G}_{is,1}^{11}(\alpha,H_{is}^{11}(\alpha),\breve{D}_{is}^{11}(\alpha),K_{is}(\alpha),p_{is}(\alpha))\triangleq-(A_{0s}+B_{is}K_{is}(\alpha))^{T}\breve{D}_{is}^{11}(\alpha,1)$$
$$-H_{is}^{11}(\alpha,1)B_{is}p_{is}(\alpha)-K_{is}^{T}(\alpha)R_{is}p_{is}(\alpha)-Q_{is}p_{is}(\alpha) \quad (80)$$

when $2\le r\le k_{is}$

$$\breve{G}_{is,r}^{11}(\alpha,H_{is}^{11}(\alpha),\breve{D}_{is}^{11}(\alpha),K_{is}(\alpha),p_{is}(\alpha))\triangleq-(A_{0s}+B_{is}K_{is}(\alpha))^{T}\breve{D}_{is}^{11}(\alpha,r)$$
$$-H_{is}^{11}(\alpha,r)B_{is}p_{is}(\alpha) \quad (81)$$

$$\breve{G}_{is,1}^{21}(\alpha,H_{is}^{21}(\alpha),\breve{D}_{is}^{11}(\alpha),\breve{D}_{is}^{21}(\alpha),p_{is}(\alpha))\triangleq-(L_{is}^{*}(\alpha)C_{is})^{T}\breve{D}_{is}^{11}(\alpha,1)$$
$$-(A_{0s}-L_{is}^{*}(\alpha)C_{is}+L_{-is}^{*}(\alpha)C_{-is})^{T}\breve{D}_{is}^{21}(\alpha,1)-H_{is}^{21}(\alpha,1)B_{is}p_{is}(\alpha)-Q_{is}p_{is}(\alpha) \quad (82)$$

when $2\le r\le k_{is}$

$$\breve{G}_{is,r}^{21}(\alpha,H_{is}^{21}(\alpha),\breve{D}_{is}^{11}(\alpha),\breve{D}_{is}^{21}(\alpha),p_{is}(\alpha))\triangleq-(L_{is}^{*}(\alpha)C_{is})^{T}\breve{D}_{is}^{11}(\alpha,r)$$
$$-(A_{0s}-L_{is}^{*}(\alpha)C_{is}+L_{-is}^{*}(\alpha)C_{-is})^{T}\breve{D}_{is}^{21}(\alpha,r)-H_{is}^{21}(\alpha,r)B_{is}p_{is}(\alpha) \quad (83)$$

$$G_{is,1}(\alpha,H_{is}^{11}(\alpha),H_{is}^{12}(\alpha),H_{is}^{21}(\alpha),H_{is}^{22}(\alpha),\breve{D}_{is}^{11}(\alpha),p_{is}(\alpha))$$
$$\triangleq-2(\breve{D}_{is}^{11}(\alpha,1))^{T}B_{is}p_{is}(\alpha)-\mathrm{Tr}\left\{H_{is}^{11}(\alpha,1)\Pi_{11}^{s}(\alpha)\right\}+\mathrm{Tr}\left\{H_{is}^{12}(\alpha,1)\Pi_{21}^{s}(\alpha)\right\}$$
$$-\mathrm{Tr}\left\{H_{is}^{21}(\alpha,1)\Pi_{12}^{s}(\alpha)\right\}+\mathrm{Tr}\left\{H_{is}^{22}(\alpha,1)\Pi_{22}^{s}(\alpha)\right\}-p_{is}^{T}(\alpha)R_{is}p_{is}(\alpha) \quad (84)$$

when $2\le r\le k_{is}$

$$G_{is,r}(\alpha,H_{is}^{11}(\alpha),H_{is}^{12}(\alpha),H_{is}^{21}(\alpha),H_{is}^{22}(\alpha),\breve{D}_{is}^{11}(\alpha),p_{is}(\alpha))$$
$$\triangleq-2(\breve{D}_{is}^{11}(\alpha,r))^{T}B_{is}p_{is}(\alpha)-\mathrm{Tr}\left\{H_{is}^{11}(\alpha,r)\Pi_{11}^{s}(\alpha)\right\}+\mathrm{Tr}\left\{H_{is}^{12}(\alpha,r)\Pi_{21}^{s}(\alpha)\right\}$$
$$-\mathrm{Tr}\left\{H_{is}^{21}(\alpha,r)\Pi_{12}^{s}(\alpha)\right\}+\mathrm{Tr}\left\{H_{is}^{22}(\alpha,r)\Pi_{22}^{s}(\alpha)\right\}. \quad (85)$$

The product system of the dynamical equations (66)–(71), whose mappings are constructed by the Cartesian products of the constituents of (72)–(85), for example, $\mathcal{F}_{is}^{11}\triangleq F_{is,1}^{11}\times\cdots\times F_{is,k_{is}}^{11}$, $\mathcal{F}_{is}^{12}\triangleq F_{is,1}^{12}\times\cdots\times F_{is,k_{is}}^{12}$, $\mathcal{F}_{is}^{21}\triangleq F_{is,1}^{21}\times\cdots\times F_{is,k_{is}}^{21}$, $\mathcal{F}_{is}^{22}\triangleq F_{is,1}^{22}\times\cdots\times F_{is,k_{is}}^{22}$, $\breve{\mathcal{G}}_{is}^{11}\triangleq\breve{G}_{is,1}^{11}\times\cdots\times\breve{G}_{is,k_{is}}^{11}$, $\breve{\mathcal{G}}_{is}^{21}\triangleq\breve{G}_{is,1}^{21}\times\cdots\times\breve{G}_{is,k_{is}}^{21}$,

and $\mathcal{G}_{is} \triangleq G_{is,1} \times \cdots \times G_{is,k_{is}}$ in the optimal statistical control with output-feedback compensation, is described by

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{is}^{11}(\alpha) = \mathcal{F}_{is}^{11}(\alpha, \mathcal{H}_{is}^{11}(\alpha), \mathcal{H}_{is}^{12}(\alpha), \mathcal{H}_{is}^{21}(\alpha), K_{is}(\alpha)), \quad \mathcal{H}_{is}^{11}(t_f) \tag{86}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{is}^{12}(\alpha) = \mathcal{F}_{is}^{12}(\alpha, \mathcal{H}_{is}^{11}(\alpha), \mathcal{H}_{is}^{12}(\alpha), \mathcal{H}_{is}^{22}(\alpha), K_{is}(\alpha)), \quad \mathcal{H}_{is}^{12}(t_f) \tag{87}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{is}^{21}(\alpha) = \mathcal{F}_{is}^{21}(\alpha, \mathcal{H}_{is}^{11}(\alpha), \mathcal{H}_{is}^{21}(\alpha), \mathcal{H}_{is}^{22}(\alpha), K_{is}(\alpha)), \quad \mathcal{H}_{is}^{21}(t_f) \tag{88}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{is}^{22}(\alpha) = \mathcal{F}_{is}^{22}(\alpha, \mathcal{H}_{is}^{12}(\alpha), \mathcal{H}_{is}^{21}(\alpha), \mathcal{H}_{is}^{22}(\alpha)), \quad \mathcal{H}_{is}^{22}(t_f) \tag{89}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\breve{\mathcal{D}}_{is}^{11}(\alpha) = \breve{\mathcal{G}}_{is}^{11}(\alpha, \mathcal{H}_{is}^{11}(\alpha), \breve{\mathcal{D}}_{is}^{11}(\alpha), K_{is}(\alpha), p_{is}(\alpha)), \quad \breve{\mathcal{D}}_{is}^{11}(t_f) \tag{90}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\breve{\mathcal{D}}_{is}^{21}(\alpha) = \breve{\mathcal{G}}_{is}^{21}(\alpha, \mathcal{H}_{is}^{21}(\alpha), \breve{\mathcal{D}}_{is}^{11}(\alpha), \breve{\mathcal{D}}_{is}^{21}(\alpha), p_{is}(\alpha)), \quad \breve{\mathcal{D}}_{is}^{21}(t_f) \tag{91}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{D}_{is}(\alpha) = \mathcal{G}_{is}(\alpha, \mathcal{H}_{is}^{11}(\alpha), \mathcal{H}_{is}^{12}(\alpha), \mathcal{H}_{is}^{21}(\alpha), \mathcal{H}_{is}^{22}(\alpha), \breve{\mathcal{D}}_{is}^{11}(\alpha), p_{is}(\alpha)), \mathcal{D}_{is}(t_f) \tag{92}$$

wherein the terminal-value conditions $\mathcal{H}_{is}^{11}(t_f) = \mathcal{H}_{is}^{12}(t_f) = \mathcal{H}_{is}^{21}(t_f) = \mathcal{H}_{is}^{22}(t_f) \triangleq \underbrace{Q_{0i}^{f} \times 0 \times \cdots \times 0}_{k_{is}\text{-times}}$, $\breve{\mathcal{D}}_{is}^{11}(t_f) = \breve{\mathcal{D}}_{is}^{21}(t_f) \triangleq \underbrace{0 \times \cdots \times 0}_{k_{is}\text{-times}}$, and $\breve{\mathcal{D}}_{is}(t_f) \triangleq \underbrace{0 \times \cdots \times 0}_{k_{is}\text{-times}}$.

As for the problem statements of the control decision optimization concerned by decision maker i, the product systems (86)–(92) of the dynamical equations (66)–(71) are now further described in terms of $\mathcal{F}_{is} \triangleq \mathcal{F}_{is}^{11} \times \mathcal{F}_{is}^{12} \times \mathcal{F}_{is}^{21} \times \mathcal{F}_{is}^{22}$ and $\breve{\mathcal{G}}_{is} \triangleq \breve{\mathcal{G}}_{is}^{11} \times \breve{\mathcal{G}}_{is}^{21}$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{H}_{is}(\alpha) = \mathcal{F}_{is}(\alpha, \mathcal{H}_{is}(\alpha), K_{is}(\alpha)), \qquad \mathcal{H}_{is}(t_f) \tag{93}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\breve{\mathcal{D}}_{is}(\alpha) = \breve{\mathcal{G}}_{is}(\alpha, \mathcal{H}_{is}(\alpha), \breve{\mathcal{D}}_{is}(\alpha), K_{is}, p_{is}(\alpha)), \qquad \breve{\mathcal{D}}_{is}(t_f) \tag{94}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{D}_{is}(\alpha) = \mathcal{G}_{is}(\alpha, \mathcal{H}_{is}(\alpha), \breve{\mathcal{D}}_{is}(\alpha), p_{is}(\alpha)), \qquad \mathcal{D}_{is}(t_f) \tag{95}$$

whereby the terminal-value conditions $\mathcal{H}_{is}(t_f) \triangleq (\mathcal{H}_{is}^{11}(t_f), \mathcal{H}_{is}^{12}(t_f), \mathcal{H}_{is}^{21}(t_f), \mathcal{H}_{is}^{22}(t_f))$, and $\breve{\mathcal{D}}_{is}(t_f) \triangleq (\breve{\mathcal{D}}_{is}^{11}(t_f), \breve{\mathcal{D}}_{is}^{21}(t_f))$.

Recall that the aim is to determine risk-bearing decision u_{is} so as to minimize the performance vulnerability of (53) against all sample-path realizations of the underlying stochastic environment w_{is}. Henceforth, performance risks are interpreted

as worries and fears about certain undesirable characteristics of performance distributions of (53) and thus are proposed to manage through a finite set of selective weights. This custom set of design freedoms representing particular uncertainty aversions decision maker i is hence different from the ones with aversion to risk captured in risk-sensitive optimal control [8,9]; just to name a few.

Definition 7 (Slow Interactions—Risk-Value Aware Performance Index). With reference to L_{is}^* and L_{-is}^* being conducted optimally, the new performance index for slow interactions; that is, $\phi_{is}^0 : \{t_0\} \times (\mathbb{R}^{n_0 \times n_0})^{k_{is}} \times (\mathbb{R}^{n_0})^{k_{is}} \times \mathbb{R}^{k_{is}} \mapsto \mathbb{R}^+$ with $k_{is} \in \mathbb{N}$ is defined as a multi-criteria objective using the first k_{is} performance-measure statistics of the integral-quadratic utility (53), on the one hand, and a value and risk model, on the other, to reflect the trade-offs between reliable attainments and risks

$$\phi_{is}^0 \triangleq \underbrace{\mu_{is}^1 \kappa_{is}^1}_{\text{Standard Measure}} + \underbrace{\mu_{is}^2 \kappa_{is}^2 + \cdots + \mu_{is}^{k_{is}} \kappa_{is}^{k_{is}}}_{\text{Risk Measures}} \tag{96}$$

where the rth performance-measure statistics $\kappa_{is}^r \equiv \kappa_{is}^r(t_0, \mathcal{H}_{is}(t_0), \breve{\mathcal{D}}_{is}(t_0), \mathcal{D}_{is}(t_0)) = x_{00}^T \mathcal{H}_{is,r}^{11}(t_0) x_{00} + 2x_{00}^T \breve{\mathcal{D}}_{is,r}^{11}(t_0) + \mathcal{D}_{is,r}(t_0)$, while the dependence of $\mathcal{H}_{is}$, $\breve{\mathcal{D}}_{is}$, and $\mathcal{D}_{is}$ on certain admissible K_{is} and p_{is} is omitted for notational simplicity. In addition, parametric design measures μ_{is}^r from the sequence $\mu^{is} = \{\mu_{is}^r \geq 0\}_{r=1}^{k_{is}}$ with $\mu_{is}^1 > 0$ concentrate on various prioritizations as chosen by decision maker i toward his/her trade-offs between performance robustness and high performance demands.

To specifically indicate the dependence of the risk-value aware performance index (96) expressed in Mayer form on u_{is} and the set of interferences from all other decision makers δ_{-is}, the multi-criteria objective (96) for decision maker i is now rewritten as $\phi_{is}^0(u_{is}; \delta_{-is})$. In view of this multiperson decision problem, a noncooperative Nash equilibrium ensures that no decision makers have incentive to unilaterally deviate from the equilibrium decisions in order to further optimize their performance. Henceforth, a Nash game-theoretic framework is suitable to capture the nature of conflicts as actions of a decision maker are tightly coupled with those of other remaining decision makers.

Definition 8 (Slow Interactions—Nash Equilibrium). An admissible set of actions $(u_{1s}^*, \ldots, u_{Ns}^*)$ is a Nash equilibrium for an N-person stochastic game where each decision maker i and $i = 1, \ldots, N$ has the performance index $\phi_{is}^0(u_{is}; \delta_{-is})$ of Mayer type, if for all admissible $(u_{1s}, \ldots, u_{Ns})$ the following inequalities hold:

$$\phi_{is}^0(u_{is}^*; \delta_{-is}^*) \leq \phi_{is}^0(u_{is}; \delta_{-is}^*), \quad i = 1, \ldots, N.$$

When solving for a Nash equilibrium solution, it is very important to realize that N decision makers have different performance indices to minimize. A standard approach for a potential solution from the set of N inequalities as stated above is

to solve jointly N optimal control decision problems defined by these inequalities, each of which depends structurally on the other decision maker's decision laws. However, a Nash equilibrium solution cannot be unique due to informational nonuniqueness. The problems with informational nonuniqueness under the feedback information pattern and the need for more satisfactory resolution have been addressed via the requirement of a Nash equilibrium solution to have an additional property that its restriction on either the final part $[t, t_f]$ or the initial part $[t_0, \varepsilon]$ is a Nash solution to the truncated version of either traditional games with terminal costs or the statistics-based games with initial costs herein, defined on either $[t, t_f]$ or $[t_0, \varepsilon]$, respectively. With such a restriction so defined, the solution is now termed as a feedback Nash equilibrium solution, which is now free of any informational nonuniqueness, and thus whose derivation allows a dynamic programming type argument.

In conformity with the rigorous formulation of dynamic programming, the following development is important. Let the terminal time t_f and states $(\mathcal{H}_{is}(t_f), \breve{\mathcal{D}}_{is}(t_f), \mathcal{D}_{is}(t_f))$ be given. Then the other end condition involved the initial time t_0 and corresponding states $(\mathcal{H}_{is}(t_0), \breve{\mathcal{D}}_{is}(t_0), \mathcal{D}_{is}(t_0))$ are specified by a target set requirement.

Definition 9 (Slow Interactions—Target Sets). $(t_0, \mathcal{H}_{is}(t_0), \breve{\mathcal{D}}_{is}(t_0), \mathcal{D}_{is}(t_0)) \in \hat{\mathcal{M}}_{is}$ where the target set $\hat{\mathcal{M}}_{is}$ and $i = 1, \ldots, N$ is a closed subset of $\{t_0\} \times (\mathbb{R}^{n_0 \times n_0})^{4k_{is}} \times (\mathbb{R}^{n_0})^{2k_{is}} \times \mathbb{R}^{k_{is}}$.

For the given terminal data $(t_f, \mathcal{H}_{is}(t_f), \breve{\mathcal{D}}_{is}(t_f), \mathcal{D}_{is}(t_f))$ wherein $\mathcal{H}_{is}^f \triangleq \mathcal{H}_{is}(t_f)$, $\breve{\mathcal{D}}_{is}^f \triangleq \breve{\mathcal{D}}_{is}(t_f)$, and $\mathcal{D}_{is}^f \triangleq \mathcal{D}_{is}(t_f)$, the classes $\hat{\mathcal{K}}^{is}_{t_f, \mathcal{H}_{is}^f, \breve{\mathcal{D}}_{is}^f, \mathcal{D}_{is}^f; \mu^{is}}$ and $\hat{\mathcal{P}}^{is}_{t_f, \mathcal{H}_{is}^f, \breve{\mathcal{D}}_{is}^f, \mathcal{D}_{is}^f; \mu^{is}}$ of admissible feedback decisions are now defined as follows.

Definition 10 (Slow Interactions—Admissible Feedback Sets). Let the compact subset $\overline{\mathcal{K}}_{is} \subset \mathbb{R}^{m_i \times n_0}$ and $\overline{\mathcal{P}}_{is} \subset \mathbb{R}^{m_i}$ be the respective sets of allowable values. For the given $k_{is} \in \mathbb{N}$ and the sequence $\mu^{is} = \{\mu_{is}^r \geq 0\}_{r=1}^{k_{is}}$ with $\mu_{is}^1 > 0$, let $\hat{\mathcal{K}}^{is}_{t_f, \mathcal{H}_{is}^f, \breve{\mathcal{D}}_{is}^f, \mathcal{D}_{is}^f; \mu^{is}}$ and $\hat{\mathcal{P}}^{is}_{t_f, \mathcal{H}_{is}^f, \breve{\mathcal{D}}_{is}^f, \mathcal{D}_{is}^f; \mu^{is}}$ be the classes of $\mathcal{C}([t_0, t_f]; \mathbb{R}^{m_i \times n_0})$ and $\mathcal{C}([t_0, t_f]; \mathbb{R}^{m_i})$ with values $K_{is}(\cdot) \in \overline{\mathcal{K}}_{is}$ and $p_{is}(\cdot) \in \overline{\mathcal{P}}_{is}$, for which the risk-value aware performance index (23) is finite and the trajectory solutions to the dynamic equations (93)–(95) reach $(t_0, \mathcal{H}_{is}(t_0), \breve{\mathcal{D}}_{is}(t_0), \mathcal{D}_{is}(t_0)) \in \hat{\mathcal{M}}_{is}$ and $i = 1, \ldots, N$.

In the sequel, when decision maker i is confident that other $N - 1$ decision makers choose their feedback Nash equilibrium strategies, that is, (K_{1s}^*, p_{1s}^*), $\ldots$, $(K_{(i-1)s}^*, p_{(i-1)s}^*)$, $(K_{(i+1)s}^*, p_{(i+1)s}^*)$, $\ldots$, (K_{Ns}^*, p_{Ns}^*). He/she then uses his/her feedback Nash equilibrium strategy (K_{is}^*, p_{is}^*).

Definition 11 (Slow Interactions—Feedback Nash Equilibrium). Let $u_{is}^*(t) = K_{is}^*(t)\hat{x}_{is}(t) + p_{is}^*(t)$, or equivalently (K_{is}^*, p_{is}^*) constitute a feedback Nash equilibrium such that

$$\phi_{is}^0(K_{is}^*, p_{is}^*; \delta_{-is}^*) \leq \phi_{is}^0(K_{is}, p_{is}; \delta_{-is}^*), \quad i = 1, \ldots, N \tag{97}$$

for all admissible $K_{is} \in \hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$ and $p_{is} \in \hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$, upon which the solutions to the dynamical systems (93)–(95) exist on $[t_0, t_f]$.

Then, $((K^*_{1s}, p^*_{1s}), \ldots, (K^*_{Ns}, p^*_{Ns}))$ when restricted to the interval $[t_0, \alpha]$ is still a feedback Nash equilibrium for the set of Nash control and decision problems with the appropriate terminal-value conditions $(\alpha, \mathcal{H}^*_{is}(\alpha), \breve{\mathcal{D}}^*_{is}(\alpha), \mathcal{D}^*_{is}(\alpha))$ for all $\alpha \in [t_0, t_f]$.

Now, the decision optimization residing at decision maker i is to minimize the risk-value aware performance index (96) for all admissible $K_{is} \in \hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$ and $p_{is} \in \hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$ while subject to interferences from all remaining decision makers δ^*_{-is}.

Definition 12 (Slow Interactions—Optimization of Mayer Problem). Assume that there exist $k_{is} \in \mathbb{N}$, $i = 1, \ldots, N$, and the sequence of nonnegative scalars $\mu^{is} = \{\mu^r_{is} \geq 0\}_{r=1}^{k_{is}}$ with $\mu^1_{is} > 0$. Then, the decision optimization for decision maker i over $[t_0, t_f]$ is given by

$$\min_{K_{is} \in \hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}, p_{is} \in \hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}} \phi^0_{is}(K_{is}, p_{is}; \delta^*_{-is}) \tag{98}$$

subject to the dynamic equations (93)–(95), for $\alpha \in [t_0, t_f]$.

Notice that the optimization considered here is in Mayer form and can be solved by applying an adaptation of the Mayer form verification theorem of dynamic programming given in [5]. To embed the aforementioned optimization into a larger optimization problem, the terminal time and states $(t_f, \mathcal{H}_{is}(t_f), \breve{\mathcal{D}}_{is}(t_f), \mathcal{D}_{is}(t_f))$ are parameterized as $(\varepsilon, \mathcal{Y}_{is}, \breve{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ whereby $\mathcal{Y}_{is} \triangleq \mathcal{H}_{is}(\varepsilon)$, $\breve{\mathcal{Z}}_{is} \triangleq \breve{\mathcal{D}}_{is}(\varepsilon)$, and $\mathcal{Z}_{is} \triangleq \mathcal{D}_{is}(\varepsilon)$. Thus, the value function for this optimization problem is now depending on parameterizations of terminal-value conditions.

Definition 13 (Slow Interactions—Value Function). Let $(\varepsilon, \mathcal{Y}_{is}, \breve{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) \in [t_0, t_f] \times (\mathbb{R}^{n_0 \times n_0})^{4k_{is}} \times (\mathbb{R}^{n_0})^{2k_{is}} \times \mathbb{R}^{k_{is}}$ be given. Then, the value function $\mathcal{V}_{is}(\varepsilon, \mathcal{Y}_{is}, \breve{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ associated with decision maker i and $i = 1, \ldots, N$ is defined by

$$\mathcal{V}_{is}(\varepsilon, \mathcal{Y}_{is}, \breve{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) = \inf_{K_{is} \in \hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}, p_{is} \in \hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}} \phi^0_{is}(K_{is}, p_{is}; \delta^*_{-is}). \tag{99}$$

It is conventional to let $\mathcal{V}_{is}(\varepsilon, \mathcal{Y}_{is}, \breve{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) = +\infty$ when either $\hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$ or $\hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$ is empty.

Unless otherwise specified, the dependence of trajectory solutions $\mathcal{H}_{is}(\cdot)$, $\breve{\mathcal{D}}_{is}(\cdot)$, and $\mathcal{D}_{is}(\cdot)$ on $(K_{is}, p_{is}; \delta^*_{-is})$ is now omitted for notational clarity. The results that

follow summarize some properties of the value function as necessary conditions for optimality whose verifications can be obtained via parallel adaptations [6] to those of excellent treatments in [5].

Theorem 8 (Slow Interactions—Necessary Conditions). *The value function associated with decision maker i and $i = 1, \ldots, N$ evaluated along any time-backward trajectory corresponding to a feedback decision feasible for its terminal states is an increasing function of time. Moreover, the value function evaluated along any optimal time-backward trajectory is constant.*

As far as a construction of scalar-valued functions $\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$, which then serve as potential candidates for the value function, is concerned, these necessary conditions are also sufficient for optimality as shown in the next result.

Theorem 9 (Slow Interactions—Sufficient Condition). *Let $\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ be an extended real-valued function on $[t_0, t_f] \times (\mathbb{R}^{n_0 \times n_0})^{4k_{is}} \times (\mathbb{R}^{n_0})^{2k_{is}} \times \mathbb{R}^{k_{is}}$ such that $\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) \equiv \phi^0_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}; \delta^*_{-is})$ for decision maker i and $i = 1, \ldots, N$. Further, let t_f, $\mathcal{H}^f_{is}$, $\check{\mathcal{D}}^f_{is}$, and $\mathcal{D}^f_{is}$ be given the terminal-value conditions. Suppose, for each trajectory $(\mathcal{H}_{is}, \check{\mathcal{D}}_{is}, \mathcal{D}_{is})$ corresponding to a permissible decision strategy (K_{is}, p_{is}) in $\hat{\mathcal{K}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}}$ and $\hat{\mathcal{P}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}}$, that $\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ is finite and time-backward increasing on $\left[t_0, t_f\right]$.*

If (K^*_{is}, p^*_{is}) is a permissible strategy in $\hat{\mathcal{K}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}}$ and $\hat{\mathcal{P}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}}$ such that for the corresponding trajectory $(\mathcal{H}^*_{is}, \check{\mathcal{D}}^*_{is}, \mathcal{D}^*_{is})$, $\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ is constant then (K^*_{is}, p^*_{is}) is a feedback Nash strategy. Therefore, $\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) \equiv \mathcal{V}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$.

Proof. Given the space limitation, the detailed analysis and development are now referred to the work by the first author [6].

Definition 14 (Slow Interactions—Reachable Sets). Let reachable set $\{\hat{\mathcal{Q}}_{is}\}_{i=1}^N$ for decision maker i be defined as follows

$$\hat{\mathcal{Q}}_{is} \triangleq \Big\{(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) \in [t_0, t_f] \times (\mathbb{R}^{n_0 \times n_0})^{4k_{is}} \times (\mathbb{R}^{n_0})^{2k_{is}} \times \mathbb{R}^{k_{is}}$$
$$\text{such that } \hat{\mathcal{K}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}} \neq \emptyset \text{ and } \hat{\mathcal{P}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}} \neq \emptyset\Big\}.$$

Moreover, it can be shown that the value function associated with decision maker i is satisfying a partial differential equation at each interior point of $\hat{\mathcal{Q}}_{is}$ at which it is differentiable.

Theorem 10 (Slow Interactions—Hamilton–Jacobi–Bellman (HJB) Equation). *Let $(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ be any interior point of the reachable set $\hat{\mathcal{Q}}_{is}$, at which the value function $\mathcal{V}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ is differentiable. If there exists a feedback Nash equilibrium $(K^*_{is}, p^*_{is}) \in \hat{\mathcal{K}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}} \times \hat{\mathcal{P}}^{is}_{t_f, \mathcal{H}^f_{is}, \check{\mathcal{D}}^f_{is}, \mathcal{D}^f_{is}; \mu^{is}}$, then the differential equation*

$$
\begin{aligned}
0 = \min_{(K_{is},p_{is})\in\overline{\mathcal{K}}_{if}\times\overline{\mathcal{P}}_{is}} \Bigg\{ & \frac{\partial}{\partial\varepsilon}\mathcal{V}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is}) \\
& + \frac{\partial}{\partial\,\mathrm{vec}(\mathcal{Y}_{is})}\mathcal{V}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is})\mathrm{vec}(\mathcal{F}_{is}(\varepsilon,\mathcal{Y}_{is},K_{is})) \\
& + \frac{\partial}{\partial\,\mathrm{vec}(\breve{\mathcal{Z}}_{is})}\mathcal{V}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is})\mathrm{vec}(\breve{\mathcal{G}}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},K_{is},p_{is})) \\
& + \frac{\partial}{\partial\,\mathrm{vec}(\mathcal{Z}_{is})}\mathcal{V}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is})\mathrm{vec}(\mathcal{G}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},p_{is}))\Bigg\}
\end{aligned} \tag{100}
$$

is satisfied where the boundary condition $\mathcal{V}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is}) = \phi^0_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is})$.

Proof. By what have been shown in the recent results by the first author [6], the detailed development for the result herein can be easily proven.

Finally, the following result gives the sufficient condition used to verify a feedback Nash strategy for decision maker i and $i = 1,\ldots,N$.

Theorem 11 (Slow Interactions—Verification Theorem). *Let* $\mathcal{W}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is})$ *and* $i = 1,\ldots,N$ *be continuously differentiable solution of the HJB equation (100) which satisfies the boundary condition*

$$
\mathcal{W}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is}) = \phi^0_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is}) \ . \tag{101}
$$

Let $(t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is}) \in \hat{\mathcal{Q}}_{is}$; $(K^*_{is},p^*_{is}) \in \hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}} \times \hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$; *and the corresponding solutions* $(\mathcal{H}_{is},\breve{\mathcal{D}}_{is},\mathcal{D}_{is})$ *of the dynamical equations (93)–(95). Then,* $\mathcal{W}_{is}(\varepsilon,\mathcal{Y}_{is},\breve{\mathcal{Z}}_{is},\mathcal{Z}_{is})$ *is time-backward increasing function of* α.

If (K^*_{is},p^*_{is}) *is in* $\hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}} \times \hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}^f_{is},\breve{\mathcal{D}}^f_{is},\mathcal{D}^f_{is};\mu^{is}}$ *defined on* $[t_0,t_f]$ *with the corresponding solutions* $(\mathcal{H}^*_{is},\breve{\mathcal{D}}^*_{is},\mathcal{D}^*_{is})$ *of the dynamical equations (93)–(95) such that, for* $\alpha \in [t_0,t_f]$

$$
\begin{aligned}
0 = & \frac{\partial}{\partial\varepsilon}\mathcal{W}_{is}(\alpha,\mathcal{H}^*_{is}(\alpha),\breve{\mathcal{D}}^*_{is}(\alpha),\mathcal{D}^*_{is}(\alpha)) \\
& + \frac{\partial}{\partial\,\mathrm{vec}(\mathcal{Y}_{is})}\mathcal{W}_{is}(\alpha,\mathcal{H}^*_{is}(\alpha),\breve{\mathcal{D}}^*_{is}(\alpha),\mathcal{D}^*_{is}(\alpha))\mathrm{vec}(\mathcal{F}_{is}(\alpha,\mathcal{H}^*_{is}(\alpha),K^*_{is}(\alpha))) \\
& + \frac{\partial}{\partial\,\mathrm{vec}(\breve{\mathcal{Z}}_{is})}\mathcal{W}_{is}(\alpha,\mathcal{H}^*_{is}(\alpha),\breve{\mathcal{D}}^*_{is}(\alpha),\mathcal{D}^*_{is}(\alpha))\mathrm{vec}(\breve{\mathcal{G}}_{is}(\alpha,\mathcal{H}^*_{is}(\alpha),\breve{\mathcal{D}}^*_{is}, \\
& \times K^*_{is}(\alpha),p^*_{is}(\alpha))) \\
& + \frac{\partial}{\partial\,\mathrm{vec}(\mathcal{Z}_{is})}\mathcal{W}_{is}(\alpha,\mathcal{H}^*_{is}(\alpha),\breve{\mathcal{D}}^*_{is}(\alpha),\mathcal{D}^*_{is}(\alpha))\mathrm{vec}(\mathcal{G}_{is}(\alpha,\mathcal{H}^*_{is}(\alpha), \\
& \times \breve{\mathcal{D}}^*_{is}(\alpha),p^*_{is}(\alpha)))
\end{aligned} \tag{102}
$$

then, (K_{is}^*, p_{is}^*) *is a feedback Nash strategy in* $\hat{\mathcal{K}}^{is}_{t_f,\mathcal{H}_{is}^f,\check{\mathcal{D}}_{is}^f,\mathcal{D}_{is}^f;\mu^{is}} \times \hat{\mathcal{P}}^{is}_{t_f,\mathcal{H}_{is}^f,\check{\mathcal{D}}_{is}^f,\mathcal{D}_{is}^f;\mu^{is}}$,

$$\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) = \mathcal{V}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) \tag{103}$$

where $\mathcal{V}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ *is the value function associated with decision maker* i.

Proof. With the aid of the recent development [6], the proof then follows for the verification theorem herein.

Regarding the Mayer-type optimization problem herein, it can be solved by applying an adaptation of the Mayer form verification theorem of dynamic programming as in (102). Therefore, the terminal time and states $(\varepsilon, \mathcal{H}_{is}^f, \check{\mathcal{D}}_{is}^f, \mathcal{D}_{is})$ of the dynamics (93)–(95) are now parameterized as $(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ for a broader family of optimization problems. To apply properly the dynamic programming approach based on the HJB mechanism, together with the verification result, the solution procedure should be formulated as follows. For any given interior point $(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ of the reachable set $\hat{\mathcal{Q}}_{is}$ and $i = 1, \ldots, N$, at which the following real-valued function is considered as a candidate solution $\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ to the HJB equation (100).

Because the initial state x_{00}, which is arbitrarily fixed represents both quadratic and linear contributions to the performance index (96) of Mayer type, it hence leads to suspect that the value function is linear and quadratic in x_{00}. Thus, a candidate function $\mathcal{W}_{is} \in \mathcal{C}^1(t_0, t_f; \mathbb{R})$ for the value function is expected to have the form

$$\begin{aligned}\mathcal{W}_{is}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) = {} & x_{00}^T \sum_{r=1}^{k_{is}} \mu_{is}^r (\mathcal{Y}_{is,r}^{11} + \mathcal{E}_{is}^r(\varepsilon)) x_{00} \\ & + 2x_{00}^T \sum_{r=1}^{k_{is}} \mu_{is}^r (\check{\mathcal{Z}}_{is,r}^{11} + \check{\mathcal{T}}_{is}^r(\varepsilon)) + \sum_{r=1}^{k_{is}} \mu_{is}^r (\mathcal{Z}_{is,r} + \mathcal{T}_{is}^r(\varepsilon))\end{aligned} \tag{104}$$

where the parametric functions of time $\mathcal{E}_{is}^r \in \mathcal{C}^1(t_0, t_f; \mathbb{R}^{n_0 \times n_0})$, $\check{\mathcal{T}}_{is}^r \in \mathcal{C}^1(t_0, t_f; \mathbb{R}^{n_0})$, and $\mathcal{T}_{is}^r \in \mathcal{C}^1(t_0, t_f; \mathbb{R})$ are yet to be determined.

Moreover, it can be shown that the derivative of $\mathcal{W}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is})$ with respect to time ε is

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{W}(\varepsilon, \mathcal{Y}_{is}, \check{\mathcal{Z}}_{is}, \mathcal{Z}_{is}) = {} & x_{00}^T \sum_{r=1}^{k_{is}} \mu_{is}^r \big(\mathcal{F}_{is,r}^{11}(\varepsilon, \mathcal{Y}_{is}^{11}, \mathcal{Y}_{is}^{12}, \mathcal{Y}_{is}^{21}, K_{is}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{E}_{is}^r(\varepsilon)\big) x_{00} \\ & + 2x_{00}^T \sum_{r=1}^{k_{is}} \mu_{is}^r \big(\check{\mathcal{G}}_{is,r}^{11}(\varepsilon, \mathcal{Y}_{is}^{11}, \check{\mathcal{Z}}_{is}^{11}, K_{is}, p_{is}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\check{\mathcal{T}}_{is}^r(\varepsilon)\big) \\ & + \sum_{r=1}^{k_{is}} \mu_{is}^r \big(\mathcal{G}_{is,r}(\varepsilon, \mathcal{Y}_{is}^{11}, \mathcal{Y}_{is}^{12}, \mathcal{Y}_{is}^{21}, \mathcal{Y}_{is}^{22}, \check{\mathcal{Z}}_{is}^{11}, p_{is}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{T}_{is}^r(\varepsilon)\big).\end{aligned} \tag{105}$$

The substitution of this candidate (104) for the value function into the HJB equation (100) and making use of (105) yield

$$0 = \min_{(K_{is},p_{is})\in\overline{\mathcal{K}}_{is}\times\overline{\mathcal{P}}_{is}} \Bigg\{ x_{00}^T \sum_{r=1}^{k_{is}} \mu_{is}^r \big(\mathcal{F}_{is,r}^{11}(\varepsilon, \mathcal{Y}_{is}^{11}, \mathcal{Y}_{is}^{12}, \mathcal{Y}_{is}^{21}, K_{is}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{E}_{is}^r(\varepsilon)\big) x_{00}$$
$$+2x_{00}^T \sum_{r=1}^{k_{is}} \mu_{is}^r \big(\breve{\mathcal{G}}_{is,r}^{11}(\varepsilon, \mathcal{Y}_{is}^{11}, \breve{\mathcal{Z}}_{is}^{11}, K_{is}, p_{is}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\breve{\mathcal{T}}_{is}^r(\varepsilon)\big)$$
$$+ \sum_{r=1}^{k_{is}} \mu_{is}^r \big(\mathcal{G}_{is,r}(\varepsilon, \mathcal{Y}_{is}^{11}, \mathcal{Y}_{is}^{12}, \mathcal{Y}_{is}^{21}, \mathcal{Y}_{is}^{22}, \breve{\mathcal{Z}}_{is}^{11}, p_{is}) + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{T}_{is}^r(\varepsilon)\big)\Bigg\}. \tag{106}$$

Taking the gradient with respect to K_{is} and p_{is} of the expression within the bracket of (106) yield the necessary conditions for an extremum of risk-value performance index (96) on the time interval $[t_0, \varepsilon]$

$$K_{is} = -R_{is}^{-1}\left[B_{is}^T \sum_{r=1}^{k_{is}} \hat{\mu}_{is}^r \mathcal{Y}_{is,r}^{11} + \frac{1}{\mu_{is}^1} Q_{is}(\varepsilon)\right] \tag{107}$$

$$p_{is} = -R_{is}^{-1} B_{is}^T \sum_{r=1}^{k_{is}} \hat{\mu}_{is}^r \breve{\mathcal{Z}}_{is,r}^{11}. \tag{108}$$

where the normalized weights $\hat{\mu}_{is}^r \triangleq \frac{\mu_{is}^r}{\mu_{is}^1}$.

Given that the feedback Nash strategy (107) and (108) is applied to the expression (106), the minimum of (106) for any $\varepsilon \in [t_0, t_f]$ and when $\mathcal{Y}_{is}$, $\breve{\mathcal{Z}}_{is}$, and $\mathcal{Z}_{is}$ evaluated along the solutions to the dynamical equations (93)–(95) must be sought in the next step. As it turns out, the time-dependent functions $\{\mathcal{E}_{is}^r(\cdot)\}_{r=1}^{k_{is}}$, $\{\breve{\mathcal{T}}_{is}^r(\cdot)\}_{r=1}^{k_{is}}$, and $\{\mathcal{T}_{is}^r(\cdot)\}_{r=1}^{k_{is}}$, which will render the left-hand side of (106) equal to zero, must satisfy the time-backward differential equations, for $1 \le r \le k_{is}$

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{E}_{is}^r(\varepsilon) = -\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,r}^{11}(\varepsilon);\ \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\breve{\mathcal{T}}_{is}^r(\varepsilon) = -\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\breve{\mathcal{D}}_{is,r}^{11}(\varepsilon);\ \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{T}_{is}^r(\varepsilon) = -\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{D}_{is,r}(\varepsilon) \tag{109}$$

whereby the respective $\mathcal{H}_{is,r}^{11}(\cdot)$, $\breve{\mathcal{D}}_{is,r}^{11}(\cdot)$, and $\mathcal{D}_{is,1}(\cdot)$ are the solutions to: the backward-in-time matrix-valued differential equations

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,1}^{11}(\varepsilon) = -(A_{0s} + B_{is}K_{is}(\varepsilon))^T \mathcal{H}_{is,1}^{11}(\varepsilon) - \mathcal{H}_{is,1}^{11}(\varepsilon)(A_{0s} + B_{is}K_{is}(\varepsilon))$$
$$-Q_{0is} - K_{is}^T(\varepsilon) R_{is} K_{is}(\varepsilon) - 2Q_{is}K_{is}(\varepsilon) \tag{110}$$

when $2 \le r \le k_{is}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,r}^{11}(\varepsilon) = &-(A_{0s} + B_{is}K_{is}(\varepsilon))^T\mathcal{H}_{is,r}^{11}(\varepsilon) - \mathcal{H}_{is,r}^{11}(\varepsilon)(A_{0s} + B_{is}K_{is}(\varepsilon)) \\ &- \sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{11}(\varepsilon)\Pi_{11}^{s}(\varepsilon) + \mathcal{H}_{is,v}^{12}(\varepsilon)\Pi_{21}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{11}(\varepsilon) \\ &- \sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{11}(\varepsilon)\Pi_{12}^{s}(\varepsilon) + \mathcal{H}_{is,v}^{12}(\varepsilon)\Pi_{22}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{21}(\varepsilon)\end{aligned} \tag{111}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,1}^{12}(\varepsilon) = &-(A_{0s} + B_{is}K_{is}(\varepsilon))^T\mathcal{H}_{is,1}^{12}(\varepsilon) - \mathcal{H}_{is,1}^{11}(\varepsilon)(L_{is}^{*}(\varepsilon)C_{is}) \\ &-\mathcal{H}_{is,1}^{12}(\varepsilon)(A_{0s} - L_{is}^{*}(\varepsilon)C_{is} + L_{-is}^{*}(\varepsilon)C_{-is}) - Q_{0is}\end{aligned} \tag{112}$$

when $2 \le r \le k_{is}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,r}^{12}(\varepsilon) = &-(A_{0s} + B_{is}K_{is}(\varepsilon))^T\mathcal{H}_{is,r}^{12}(\varepsilon) \\ &-\mathcal{H}_{is,r}^{12}(\varepsilon)(A_{0s} - L_{is}^{*}(\varepsilon)C_{is} + L_{-is}^{*}(\varepsilon)C_{-is}) - \mathcal{H}_{is,r}^{11}(\varepsilon)(L_{is}^{*}(\varepsilon)C_{is}) \\ &- \sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{11}(\varepsilon)\Pi_{11}^{s}(\varepsilon) + \mathcal{H}_{is,v}^{12}(\varepsilon)\Pi_{21}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{12}(\varepsilon) \\ &- \sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{11}(\varepsilon)\Pi_{12}^{s}(\varepsilon) + \mathcal{H}_{is,v}^{12}(\varepsilon)\Pi_{22}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{22}(\varepsilon)\end{aligned} \tag{113}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,1}^{21}(\varepsilon) = &-(A_{0s} - L_{is}^{*}(\varepsilon)C_{is} + L_{-is}^{*}(\varepsilon)C_{-is})^T\mathcal{H}_{is,1}^{21}(\varepsilon) \\ &-\mathcal{H}_{is,1}^{21}(\varepsilon)(A_{0s} + B_{is}K_{is}(\varepsilon)) \\ &-Q_{0is} - 2Q_{is}K_{is}(\varepsilon) - (L_{is}^{*}(\varepsilon)C_{is})^T\mathcal{H}_{is,1}^{11}(\varepsilon)\end{aligned} \tag{114}$$

when $2 \le r \le k_{is}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,r}^{21}(\varepsilon) = &-\mathcal{H}_{is,r}^{21}(\varepsilon)(A_{0s} + B_{is}K_{is}(\varepsilon)) - (A_{0s} - L_{is}^{*}(\varepsilon)C_{is} \\ &+L_{-is}^{*}(\varepsilon)C_{-is})^T\mathcal{H}_{is,r}^{21}(\varepsilon) - (L_{is}^{*}(\varepsilon)C_{is})^T\mathcal{H}_{is,r}^{11}(\varepsilon) \\ &- \sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{21}(\varepsilon)\Pi_{11}^{s}(\varepsilon) + \mathcal{H}_{is,v}^{22}(\varepsilon)\Pi_{21}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{11}(\varepsilon)\end{aligned}$$

$$-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{21}(\varepsilon)\Pi_{12}^{s}(\varepsilon)+\mathcal{H}_{is,v}^{22}(\varepsilon)\Pi_{22}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{21}(\varepsilon) \tag{115}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,1}^{22}(\varepsilon) &= -(L_{is}^{*}(\varepsilon)C_{is})^{T}\mathcal{H}_{is,1}^{12}(\varepsilon)-(A_{0s}-L_{is}^{*}(\varepsilon)C_{is}\\ &\quad +L_{-is}^{*}(\varepsilon)C_{-is})^{T}\mathcal{H}_{is,1}^{22}(\varepsilon)-\mathcal{H}_{is,1}^{22}(\varepsilon)(A_{0s}-L_{is}^{*}(\varepsilon)C_{is}\\ &\quad +L_{-is}^{*}(\varepsilon)C_{-is})-\mathcal{H}_{is,1}^{21}(\varepsilon)(L_{is}^{*}(\varepsilon)C_{is})-Q_{0is}\end{aligned} \tag{116}$$

when $2\le r\le k_{is}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{H}_{is,r}^{22}(\varepsilon) &= -(L_{is}^{*}(\varepsilon)C_{is})^{T}\mathcal{H}_{is,r}^{12}(\varepsilon)-(A_{0s}-L_{is}^{*}(\varepsilon)C_{is}+L_{-is}^{*}(\varepsilon)C_{-is})^{T}\mathcal{H}_{is,r}^{22}(\varepsilon)\\ &\quad -\mathcal{H}_{is,r}^{22}(\varepsilon)(A_{0s}-L_{is}^{*}(\varepsilon)C_{is}+L_{-is}^{*}(\varepsilon)C_{-is})-\mathcal{H}_{is,r}^{21}(\varepsilon)(L_{is}^{*}(\varepsilon)C_{is})\\ &\quad -\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{21}(\varepsilon)\Pi_{11}^{s}(\varepsilon)+\mathcal{H}_{is,v}^{22}(\varepsilon)\Pi_{21}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{12}(\varepsilon)\\ &\quad -\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[\mathcal{H}_{is,v}^{21}(\varepsilon)\Pi_{12}^{s}(\varepsilon)+\mathcal{H}_{is,v}^{22}(\varepsilon)\Pi_{22}^{s}(\varepsilon)\right]\mathcal{H}_{is,r-v}^{22}(\varepsilon)\end{aligned} \tag{117}$$

the backward-in-time vector-valued differential equations

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\breve{\mathcal{D}}_{is,1}^{11}(\varepsilon) &= -(A_{0s}+B_{is}K_{is}(\varepsilon))^{T}\breve{\mathcal{D}}_{is,1}^{11}(\varepsilon)\\ &\quad -\mathcal{H}_{is,1}^{11}(\varepsilon)B_{is}p_{is}(\varepsilon)-K_{is}^{T}(\varepsilon)R_{is}p_{is}(\varepsilon)-Q_{is}p_{is}(\varepsilon)\end{aligned} \tag{118}$$

when $2\le r\le k_{is}$

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\breve{\mathcal{D}}_{is,r}^{11}(\varepsilon)=-(A_{0s}+B_{is}K_{is}(\varepsilon))^{T}\breve{\mathcal{D}}_{is,r}^{11}(\varepsilon)-\mathcal{H}_{is,r}^{11}(\varepsilon)B_{is}p_{is}(\varepsilon) \tag{119}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\breve{\mathcal{D}}_{is,1}^{21}(\varepsilon) &= -(A_{0s}-L_{is}^{*}(\varepsilon)C_{is}+L_{-is}^{*}(\varepsilon)C_{-is})^{T}\breve{\mathcal{D}}_{is,1}^{21}(\varepsilon)-(L_{is}^{*}(\varepsilon)C_{is})^{T}\breve{\mathcal{D}}_{is,1}^{11}(\varepsilon)\\ &\quad -\mathcal{H}_{is,1}^{21}(\varepsilon)B_{is}p_{is}(\varepsilon)-Q_{is}p_{is}(\varepsilon)\end{aligned} \tag{120}$$

when $2\le r\le k_{is}$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\breve{\mathcal{D}}_{is,r}^{21}(\varepsilon) &= -(A_{0s}-L_{is}^{*}(\varepsilon)C_{is}+L_{-is}^{*}(\varepsilon)C_{-is})^{T}\breve{\mathcal{D}}_{is,r}^{21}(\varepsilon)-(L_{is}^{*}(\varepsilon)C_{is})^{T}\breve{\mathcal{D}}_{is,r}^{11}(\varepsilon)\\ &\quad -\mathcal{H}_{is,r}^{21}(\varepsilon)B_{is}p_{is}(\varepsilon)\end{aligned} \tag{121}$$

and the backward-in-time scalar-valued differential equations

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{D}_{is,1}(\varepsilon) = -2(\breve{\mathcal{D}}^{11}_{is,1}(\varepsilon))^T B_{is}\, p_{is}(\varepsilon) - \mathrm{Tr}\left\{\mathcal{H}^{11}_{is,1}(\varepsilon)\Pi^s_{11}(\varepsilon)\right\} + \mathrm{Tr}\left\{\mathcal{H}^{12}_{is,1}(\varepsilon)\Pi^s_{21}(\varepsilon)\right\}$$
$$-\mathrm{Tr}\left\{\mathcal{H}^{21}_{is,1}(\varepsilon)\Pi^s_{12}(\varepsilon)\right\} + \mathrm{Tr}\left\{\mathcal{H}^{22}_{is,1}(\varepsilon)\Pi^s_{22}(\varepsilon)\right\} - p^T_{is}(\varepsilon)R_{is}\, p_{is}(\varepsilon) \quad (122)$$

when $2 \leq r \leq k_{is}$

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{D}_{is,r}(\varepsilon) = -2(\breve{\mathcal{D}}^{11}_{is,r}(\varepsilon))^T B_{is}\, p_{is}(\varepsilon) - \mathrm{Tr}\left\{\mathcal{H}^{11}_{is,r}(\varepsilon)\Pi^s_{11}(\varepsilon)\right\} + \mathrm{Tr}\left\{\mathcal{H}^{12}_{is,r}(\varepsilon)\Pi^s_{21}(\varepsilon)\right\}$$
$$-\mathrm{Tr}\left\{\mathcal{H}^{21}_{is,r}(\varepsilon)\Pi^s_{12}(\varepsilon)\right\} + \mathrm{Tr}\left\{\mathcal{H}^{22}_{is,r}(\varepsilon)\Pi^s_{22}(\varepsilon)\right\}. \quad (123)$$

For the remainder of the development, the requirement for boundary condition (101) yields $\mathcal{E}^r_{is}(t_0) = 0$, $\breve{\mathcal{T}}^r_{is}(t_0) = 0$, and $\mathcal{T}^r_{is}(t_0) = 0$. Finally, the sufficient condition (100) of the verification theorem is hence satisfied so the extremizing feedback Nash strategy (107) and (108) is optimal

$$K^*_{is}(\varepsilon) = -R^{-1}_{is}\Big[B^T_{is}\sum_{r=1}^{k_{is}}\hat{\mu}^r_{is}\mathcal{H}^{11*}_{is,r}(\varepsilon) + \frac{1}{\mu^1_{is}}Q_{is}(\varepsilon)\Big] \quad (124)$$

$$p^*_{is}(\varepsilon) = -R^{-1}_{is}B^T_{is}\sum_{r=1}^{k_{is}}\hat{\mu}^r_{is}\breve{\mathcal{D}}^{11*}_{is,r}(\varepsilon). \quad (125)$$

Therefore, the subsequent result for risk-bearing decisions in slow interactions is summarized for each decision maker, who strategically selects: (a) the worst-case estimation gain L^*_{is} in presence of the group interference gain L^*_{-is} and (b) the feedback Nash decision parameters K^*_{is} and p^*_{is}.

Theorem 12 (Slow Interactions—Slow-Timescale Risk-Averse Decisions). *Consider slow interactions with the optimization problem governed by the risk-value aware performance index (96) and subject to the dynamical equations (93)–(95). Fix $k_{is} \in \mathbb{N}$ for $i = 1, \ldots, N$, and the sequence of nonnegative coefficients $\mu_{is} = \{\mu^r_{is} \geq 0\}^{k_{is}}_{r=1}$ with $\mu^1_{is} > 0$. Then, a linear feedback Nash equilibrium for slow interactions minimizing (96) is given by*

$$u^*_{is}(t) = K^*_{is}(t)\hat{x}^*_{is}(t) + p^*_{is}(t), \quad t \triangleq t_0 + t_f - \alpha, \quad \alpha \in [t_0, t_f]$$
$$K^*_{is}(\alpha) = -R^{-1}_{is}\left[B^T_{is}\sum_{r=1}^{k_{is}}\hat{\mu}^r_{is}\mathcal{H}^{11*}_{is,r}(\alpha) + \frac{1}{\mu^1_{is}}Q_{is}(\alpha)\right], \quad \hat{\mu}^r_{is} \triangleq \frac{\mu^r_{is}}{\mu^1_{is}} \quad (126)$$

$$p^*_{is}(\alpha) = -R^{-1}_{is}B^T_{is}\sum_{r=1}^{k_{is}}\hat{\mu}^r_{is}\breve{\mathcal{D}}^{11*}_{is,r}(\alpha), \quad i = 1, \ldots, N \quad (127)$$

where all the parametric design freedom through $\hat{\mu}^r_{is}$ represent the preferences toward specific summary statistical measures; for example, mean variance,

skewness, etc. chosen by decision makers for their performance reliability, while $\hat{x}_{is}^{*}(\cdot)$, $\mathcal{H}_{is,r}^{11*}(\cdot)$ *and* $\breve{\mathcal{D}}_{is,r}^{11*}(\cdot)$ *are the optimal solutions of the dynamical systems (47) and (110)–(119) when the decision policy* u_{is}^{*} *and linear feedback Nash equilibrium* (K_{is}^{*}, p_{is}^{*}) *are applied.*

Remark 5. It is observed that to have a linear feedback Nash equilibrium K_{is}^{*}, p_{is}^{*}, and $i = 1, \ldots, N$ be defined and continuous for all $\alpha \in [t_0, t_f]$, the solutions $\mathcal{H}_{is}(\alpha)$, $\breve{\mathcal{D}}_{is}(\alpha)$, and $\mathcal{D}_{is}(\alpha)$ to the (93)–(95) when evaluated at $\alpha = t_0$ must also exist. Therefore, it is necessary that $\mathcal{H}_{is}(\alpha)$, $\breve{\mathcal{D}}_{is}(\alpha)$, and $\mathcal{D}_{is}(\alpha)$ are finite for all $\alpha \in [t_0, t_f)$. Moreover, the solutions of (93)–(95) exist and are continuously differentiable in a neighborhood of t_f. In fact, these solutions can further be extended to the left of t_f as long as $\mathcal{H}_{is}(\alpha)$, $\breve{\mathcal{D}}_{is}(\alpha)$, and $\mathcal{D}_{is}(\alpha)$ remain finite. Hence, the existences of unique and continuously differentiable solutions to the (93)–(95) are certain if $\mathcal{H}_{is}(\alpha)$, $\breve{\mathcal{D}}_{is}(\alpha)$, and $\mathcal{D}_{is}(\alpha)$ are bounded for all $\alpha \in [t_0, t_f)$. As the result, the candidate value functions $\mathcal{W}_{is}(\alpha, \mathcal{H}_{is}(\alpha), \breve{\mathcal{D}}_{is}(\alpha), \mathcal{D}_{is}(\alpha))$ for $i = 1, \ldots, N$ are continuously differentiable as well.

6 Conclusions

A complex system is more than the sum of its parts, and the individual decision makers that function as complex dynamical systems can be understood only by analyzing their collective behavior. This research article shows recent advances on distributed information and decision frameworks, including singular perturbation methods for weak and strong coupling approximations in large-scale systems, optimal statistical control decision algorithms for performance reliability, mutual modeling, and minimax estimation for self-coordination, and Nash game-theoretic design protocols for global mission management enabled by local and autonomous decision makers, can be brought to bear on central problems of making assumptions about how to link different levels of dynamical complexity analysis related to the emergence, risk-bearing decisions, and dissolution of hierarchical macrostructures. The emphasis is on the application of a new generation of summary statistical measures associated with the linear-quadratic class of multiperson decision making and control problems in addition of values and risks-based performance indices that can provide a new paradigm for understanding and building distributed systems, where it is assumed that the individual decision makers are autonomous: able to control their own risk-bearing behavior in the furtherance of their own goals.

Appendix: Fast Interactions

In Theorem 1, the lack of analysis of performance uncertainty and information around a class of stochastic quadratic decision problems was addressed. The central concern was to examine what means for performance riskiness from the standpoint

of higher-order characteristics pertaining to performance sampling distributions. An effective and accurate capability for forecasting all the higher-order characteristics associated with a finite horizon integral-quadratic performance-measure has been obtained in Theorem 1. For notational simplicity, the right members of the mathematical statistics, which are now considered as the dynamical equations (16)–(20) for the optimal statistical control problem herein, were denoted by the convenient mappings with the actions:

$$F_{if,1}^{11}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{21}(\alpha), K_{if}(\alpha)) \triangleq -(A_{ii} + B_{ii}K_{if}(\alpha))^T H_{if}^{11}(\alpha, 1)$$
$$-H_{if}^{11}(\alpha, 1)(A_{ii} + B_{ii}K_{if}(\alpha)) - Q_{if} - K_{if}^T(\alpha)R_{if}K_{if}(\alpha) \tag{128}$$

and, for $2 \leq r \leq k_{if}$

$$F_{if,r}^{11}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{21}(\alpha), K_{if}(\alpha))$$
$$\triangleq -(A_{ii} + B_{ii}K_{if}(\alpha))^T H_{if}^{11}(\alpha, r) - H_{if}^{11}(\alpha, r)(A_{ii} + B_{ii}K_{if}(\alpha))$$
$$-\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!}\left[H_{if}^{11}(\alpha, v)\Pi_{11}^f(\alpha) + H_{if}^{12}(\alpha, v)\Pi_{21}^f(\alpha)\right] H_{if}^{11}(\alpha, r-v)$$
$$-\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!}\left[H_{if}^{11}(\alpha, v)\Pi_{12}^f(\alpha) + H_{if}^{12}(\alpha, v)\Pi_{22}^f(\alpha)\right] H_{if}^{21}(\alpha, r-v) \tag{129}$$

$$F_{if,1}^{12}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{22}(\alpha), K_{if}(\alpha)) \triangleq -(A_{ii} + B_{ii}K_{if}(\alpha))^T H_{if}^{12}(\alpha, 1)$$
$$-H_{if}^{11}(\alpha, 1)(L_{if}(\alpha)C_{ii}) - H_{if}^{12}(\alpha, 1)(A_{ii} - L_{if}(\alpha)C_{ii}) - Q_{if} \tag{130}$$

and, for $2 \leq r \leq k_{if}$

$$F_{if,r}^{12}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{22}(\alpha), K_{if}(\alpha)) \triangleq -(A_{ii} + B_{ii}K_{if}(\alpha))^T H_{if}^{12}(\alpha, r)$$
$$-H_{if}^{11}(\alpha, r)(L_{if}(\alpha)C_{ii}) - H_{if}^{12}(\alpha, r)(A_{ii} - L_{if}(\alpha)C_{ii})$$
$$-\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!}\left[H_{if}^{11}(\alpha, v)\Pi_{11}^f(\alpha) + H_{if}^{12}(\alpha, v)\Pi_{21}^f(\alpha)\right] H_{if}^{12}(\alpha, r-v)$$
$$-\sum_{v=1}^{r-1} \frac{2r!}{v!(r-v)!}\left[H_{if}^{11}(\alpha, v)\Pi_{12}^f(\alpha) + H_{if}^{12}(\alpha, v)\Pi_{22}^f(\alpha)\right] H_{if}^{22}(\alpha, r-v) \tag{131}$$

$$F_{if,1}^{21}(\alpha, H_{if}^{11}(\alpha), H_{if}^{12}(\alpha), H_{if}^{22}(\alpha), K_{if}(\alpha)) \triangleq -(A_{ii} - L_{if}(\alpha)C_{ii})^T H_{if}^{21}(\alpha, 1)$$

$$-H_{if}^{21}(\alpha,1)(A_{ii}+B_{ii}K_{if}(\alpha))-(L_{if}(\alpha)C_{ii})^{T}H_{if}^{11}(\alpha,1)-Q_{if} \tag{132}$$

and, for $2 \le r \le k_{if}$

$$\begin{aligned}
F_{if,r}^{21}(\alpha,H_{if}^{11}(\alpha),H_{if}^{12}(\alpha),H_{if}^{22}(\alpha),K_{if}(\alpha)) &\triangleq -(A_{ii}-L_{if}(\alpha)C_{ii})^{T}H_{if}^{21}(\alpha,r)\\
&-H_{if}^{21}(\alpha,r)(A_{ii}+B_{ii}K_{if}(\alpha))-(L_{if}(\alpha)C_{ii})^{T}H_{if}^{11}(\alpha,r)\\
&-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[H_{if}^{21}(\alpha,v)\Pi_{11}^{f}(\alpha)+H_{if}^{22}(\alpha,v)\Pi_{21}^{f}(\alpha)\right]H_{if}^{11}(\alpha,r-v)\\
&-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[H_{if}^{21}(\alpha,v)\Pi_{12}^{f}(\alpha)+H_{if}^{22}(\alpha,v)\Pi_{22}^{f}(\alpha)\right]H_{if}^{21}(\alpha,r-v)
\end{aligned} \tag{133}$$

$$\begin{aligned}
F_{if,1}^{22}(\alpha,H_{if}^{12}(\alpha),H_{if}^{21}(\alpha),H_{if}^{22}(\alpha)) &\triangleq -(A_{ii}-L_{if}(\alpha)C_{ii})^{T}H_{if}^{22}(\alpha,1)-Q_{if}\\
&-H_{if}^{22}(\alpha,1)(A_{ii}-L_{if}(\alpha)C_{ii})-(L_{if}(\alpha)C_{ii})^{T}H_{if}^{12}(\alpha,1)-H_{if}^{21}(\alpha,1)(L_{if}(\alpha)C_{ii})
\end{aligned} \tag{134}$$

and, for $2 \le r \le k_{if}$

$$\begin{aligned}
F_{if,r}^{22}(\alpha,H_{if}^{12}(\alpha),H_{if}^{21}(\alpha),H_{if}^{22}(\alpha)) &\triangleq -(A_{ii}-L_{if}(\alpha)C_{ii})^{T}H_{if}^{22}(\alpha,r)\\
&-H_{if}^{22}(\alpha,r)(A_{ii}-L_{if}(\alpha)C_{ii})-(L_{if}(\alpha)C_{ii})^{T}H_{if}^{12}(\alpha,r)\\
&-H_{if}^{21}(\alpha,r)(L_{if}(\alpha)C_{ii})\\
&-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[H_{if}^{21}(\alpha,v)\Pi_{11}^{f}(\alpha)+H_{if}^{22}(\alpha,v)\Pi_{21}^{f}(\alpha)\right]H_{if}^{12}(\alpha,r-v)\\
&-\sum_{v=1}^{r-1}\frac{2r!}{v!(r-v)!}\left[H_{if}^{21}(\alpha,v)\Pi_{12}^{f}(\alpha)+H_{if}^{22}(\alpha,v)\Pi_{22}^{f}(\alpha)\right]H_{if}^{22}(\alpha,r-v)
\end{aligned} \tag{135}$$

$$\begin{aligned}
G_{if,r}(\alpha,H_{if}^{11}(\alpha),H_{if}^{12}(\alpha),H_{if}^{12}(\alpha),H_{if}^{22}(\alpha)) &\triangleq -\mathrm{Tr}\left\{H_{if}^{11}(\alpha,r)\Pi_{11}^{f}(\alpha)\right\}\\
&-\mathrm{Tr}\left\{H_{if}^{12}(\alpha,r)\Pi_{21}^{f}(\alpha)\right\}-\mathrm{Tr}\left\{H_{if}^{21}(\alpha,r)\Pi_{12}^{f}(\alpha)\right\}-\mathrm{Tr}\left\{H_{if}^{22}(\alpha,r)\Pi_{22}^{f}(\alpha)\right\},
\end{aligned} \tag{136}$$

where the Kalman filter gain $L_{if} = P_{if}C_{ii}^{T}V_{ii}^{-1}$ and the shorthand notations $\Pi_{11}^{f} = L_{if}V_{ii}L_{if}^{T}$, $\Pi_{12}^{f} = \Pi_{21}^{f} = -\Pi_{11}^{f}$, and $\Pi_{22}^{f} = G_{i}WG_{i}^{T} + L_{if}V_{ii}L_{if}^{T}$.

References

1. Saksena, V.R., Cruz, J.B. Jr.: A multimodel approach to stochastic Nash games. Automatica **18**(3), 295–305 (1982)
2. Haddad, A.: Linear filtering of singularly perturbed systems. IEEE Trans. Automat. Contr. **21**, 515–519 (1976)
3. Khalih, H., Haddad, A., Blankenship, G.: Parameter scaling and well-posedness of stochastic singularly perturbed control systems. Proceedings of Twelfth Asilomar Conference, Pacific Grove, CA (1978)
4. Pham, K.D.: New risk-averse control paradigm for stochastic two-time-scale systems and performance robustness. J. Optim. Theory. Appl. **146**(2), 511–537 (2010)
5. Fleming, W.H., Rishel, R.W.: Deterministic and Stochastic Optimal Control. Springer, New York (1975)
6. Pham, K.D.: Performance-reliability-aided decision-making in multiperson quadratic decision games against jamming and estimation confrontations. J. Optim. Theory Appl. **149**(1), 599–629 (2011)
7. Yaesh, I., Shaked, U.: Game theory approach to optimal linear state estimation and its relation to the minimum H_∞ norm estimation. IEEE Trans. Automat. Contr. **37**, 828–831 (1992)
8. Jacobson, D.H.: Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic games. IEEE Trans. Automat. Contr. **18**, 124–131 (1973)
9. Whittle, P.: Risk Sensitive Optimal Control. Wiley, New York (1990)